

RRRRRRRRRRRR	TTTTTTTTTTTTT	PPPPPPPPPPPP	AAAAAAA	DDDDDDDDDDDD
RRRRRRRRRRRR	TTTTTTTTTTTTT	PPPPPPPPPPPP	AAAAAAA	DDDDDDDDDDDD
RRRRRRRRRRRR	TTTTTTTTTTTTT	PPPPPPPPPPPP	AAAAAAA	DDDDDDDDDDDD
RRR RRR	TTT	PPP	AAA	DDD
RRR RRR	TTT	PPP	AAA	DDD
RRR RRR	TTT	PPP	AAA	DDD
RRR RRR	TTT	PPP	AAA	DDD
RRR RRR	TTT	PPP	AAA	DDD
RRR RRR	TTT	PPP	AAA	DDD
RRR RRR	TTT	PPP	AAA	DDD
RRRRRRRRRRRR	TTT	PPPPPPPPPPPP	AAA	DDD
RRRRRRRRRRRR	TTT	PPPPPPPPPPPP	AAA	DDD
RRRRRRRRRRRR	TTT	PPPPPPPPPPPP	AAA	DDD
RRR RRR	TTT	PPP	AAAAAAAAAAAAAA	DDD
RRR RRR	TTT	PPP	AAAAAAAAAAAAAA	DDD
RRR RRR	TTT	PPP	AAAAAAAAAAAAAA	DDD
RRR RRR	TTT	PPP	AAA	DDD
RRR RRR	TTT	PPP	AAA	DDD
RRR RRR	TTT	PPP	AAA	DDD
RRR RRR	TTT	PPP	AAA	DDDDDDDDDDDD
RRR RRR	TTT	PPP	AAA	DDDDDDDDDDDD
RRR RRR	TTT	PPP	AAA	DDDDDDDDDDDD

FILE1D**RTPAD

G 14

RRRRRRRR	TTTTTTTT	PPPPPPP	AAAAAA	DDDDDDDD			
RRRRRRRR	TTTTTTTT	PPPPPPP	AAAAAA	DDDDDDDD			
RR	RR	PP	PP	AA	AA	DD	DD
RR	RR	PP	PP	AA	AA	DD	DD
RR	RR	PP	PP	AA	AA	DD	DD
RR	RR	PP	PP	AA	AA	DD	DD
RRRRRRRR	TT	PPPPPPP	AA	AA	DD	DD	
RRRRRRRR	TT	PPPPPPP	AA	AA	DD	DD	
RR	RR	PP	AAAAAAAAAA	DD	DD		
RR	RR	PP	AAAAAAAAAA	DD	DD		
RR	RR	PP	AA	AA	DD	DD	
RR	RR	PP	AA	AA	DD	DD	
RR	RR	PP	AA	AA	DD	DD	
RR	RR	PP	AA	AA	DDDDDDDD		
RR	RR	PP	AA	AA	DDDDDDDD		

(1)	113	DECLARATIONS
(1)	146	RTPAD - MAIN ROUTINE
(1)	247	INIT - INITIALIZATION OF LINK, ETC.
(2)	611	RECORD QUIT - snapshot QUIT info
(3)	629	READ ONLY DATA
(3)	662	READ WRITE DATA
(3)	856	PROTOCOL TABLE PSECTS

0000 1 .TITLE RTPAD - REMOTE TERMINAL PROGRAM
0000 2 .IDENT 'V04-000'
0000 3 .PSECT RTPAD,NOWRT
0000 4
0000 5 \$DEBUGDEF
0000 6
0000 7:
0000 8*****
0000 9*
0000 10* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 11* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 12* ALL RIGHTS RESERVED.
0000 13*
0000 14* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 15* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 16* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 17* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 18* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 19* TRANSFERRED.
0000 20*
0000 21* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 22* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 23* CORPORATION.
0000 24*
0000 25* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 26* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 27*
0000 28*
0000 29*****
0000 30
0000 31++
0000 32 FACILITY: REMOTE TERMINAL SUPPORT
0000 33
0000 34 ABSTRACT:
0000 35
0000 36 THIS PROGRAM RUNS ON A LOCAL NODE TO ALLOW A TERMINAL TO APPEAR TO
0000 37 BE LOCALLY CONNECTED TO A REMOTE NODE.
0000 38
0000 39
0000 40 ENVIRONMENT: VMS - USER MODE
0000 41
0000 42
0000 43 AUTHOR: W M CARDOZA, CREATION DATE: 20-AUG-79
0000 44
0000 45 MODIFIED BY:
0000 46
0000 47 V03-017 JLV0362 Jake VanNoy 11-JUL-1984
0000 48 Add code to signal PC that QUOTA EXCEEDED occurred at.
0000 49
0000 50 V03-016 JLV0353 Jake VanNoy 10-APR-1984
0000 51 Add support for SET HOST/DTE ttcn:
0000 52
0000 53 V03-015 JLV0324 Jake VanNoy 10-JAN-1984
0000 54 Fix bug in setting of vax to vax flag.
0000 55
0000 56 V03-014 JLV0315 Jake VanNoy 7-DEC-1983
0000 57 Add logical name RTPAD\$LOG for debug purposes.

0000 58 : Used like FALSLOG, etc.
0000 59 :
0000 60 : V03-013 JLVO295 Jake VanNoy 28-JUL-1983
0000 61 : Add looping on non-zero WRITEMQIO before exiting.
0000 62 : Add SET HOST/LOG[=filespec].
0000 63 :
0000 64 : V03-012 MHB0093 Mark Bramhall 7-Mar-1983
0000 65 : Moved MAXMSG to \$RTPADDEF.
0000 66 : Reworked code to use dynamic descriptors.
0000 67 : Changed the CTERM detection algorithm.
0000 68 : Changed handling of PSTHRU messages.
0000 69 :
0000 70 : V03-011 MHB0089 Mark Bramhall 11-Feb-1983
0000 71 : Changed LIB\$NET_CONNECT to UNSSNET_CONNECT.
0000 72 :
0000 73 : V03-010 JLVO Jake VanNoy 17-Jan-1983
0000 74 : Added hooks for CTERM protocol. Broke up into two modules -
0000 75 : RTPAD and VMSRT. VMSRT now holds all of the VMS specific
0000 76 : protocol. General cleanup of existing code and comments.
0000 77 : Add use of new CLI interface, results in loss of first
0000 78 : command feature used by RSTSRT.
0000 79 :
0000 80 : V03-009 WMC0068 Wayne Cardoza 5-Oct-1982
0000 81 : Fix the previous DCL fix.
0000 82 :
0000 83 : V03-008 WMC0067 Wayne Cardoza 15-Oct-1982
0000 84 : Fix timing problem when link error arrives before the
0000 85 : mailbox message.
0000 86 : Work with new DCL keyword support.
0000 87 :
0000 88 : V03-007 JLVO214 Jake VanNoy 6-OCT-1982
0000 89 : Add Read Verify code as implemented (more or less)
0000 90 : by Steve Long.
0000 91 :
0000 92 : V03-006 WMC0066 Wayne Cardoza 20-Aug-1982
0000 93 : Take care of errors on SENSE MODE.
0000 94 :
0000 95 : V03-005 WMC0065 Wayne Cardoza 7-Jul-1982
0000 96 : Don't issue out of band set mode QIO if no change.
0000 97 :
0000 98 : V03-004 WMC0064 Wayne Cardoza 9-Apr-1982
0000 99 : Don't try to disable ^T under MCR.
0000 100 :
0000 101 : V03-003 WMC0063 Wayne Cardoza 1-Apr-1982
0000 102 : Zero the buffer before a SENSE.
0000 103 :
0000 104 : V03-002 WMC0062 Wayne Cardoza 18-Mar-1982
0000 105 : Add TERMCHAR in DIB format back for RSX, etc.
0000 106 :
0000 107 : V03-001 WMC0061 Wayne Cardoza 15-Mar-1982
0000 108 : Send extended characteristics in configuration message.
0000 109 :
0000 110 : **
0000 111 : --

```
0000 113 .SBTTL DECLARATIONS
0000 114 : DEFAULT ADDRESSING MODE
0000 115 : DEFAULT DISPLACEMENT WORD
0000 116 :
0000 117 :
0000 118 :
0000 119 : INCLUDE FILES:
0000 120 :
0000 121 $DIBDEF      ;DIB OFFSETS
0000 122 $DESCDEF    ;DESCRIPTOR DEFINITIONS
0000 123 $DVIDEF     ;GETDVI DEFINITIONS
0000 124 $IODEF      ;I/O OP CODES & MODIFIERS
0000 125 $RDDEF      ;REMOTE DEVICE PROTOCOL
0000 126 $RTPADDEF   ;*** NEW
0000 127 $TSADef     ;*** NEW *** tsadef
0000 128 $TTYDEFS    ;TERMINAL DRIVER SYMBOLS
0000 129 :
0000 130 : EQUATED SYMBOLS:
0000 131 :
0000 132 : AST CONTROL BLOCK
0000 133 :
0000 134 AST$T_BUF = CTPSB_PRO_MSGTYPE ; DATA BUFFER STARTS HERE
0000 135 :
0000 136 REM$_FACILITY = 510          ; REM$_ code
0000 137 :
0000 138 show me,MC,MD
0000 139 $$SHR_MSGDEF NAME=REM,CODE=REMS_FACILITY,SCOPE=LOCAL, -
0000 140           MSGCODES = <<ATPC,INFO5>
0000           .IF      NDF,SHRSK SHRDEF
0000           SHR$K SHRDEF = 1
0000           $$SHRDEF
0000           $DEFINI SHR,
0000           .SAVE  LOCAL_BLOCK
0000           .NOCROSS
0000           .IIF    DIF <> <GLOBAL>,ENABLE SUPPRESSION
0000           .PSECT $ABSS,ABS
0000           $GBLINI
0000           .IF    IDN <LOCAL> <GLOBAL>
0000           .MACRO $DEF  SYM,ALLOC,SIZ
0000           .IIF  NB,SYM, SYM:::
0000           .IIF  NB,ALLOC,      ALLOC SIZ
0000           .ENDM $DEF
0000           .MACRO $SEQU  SYM,VAL
0000           SYM==VAL
0000           .ENDM $SEQU
0000           .MACRO $VIELD1 MOD,SEP,SYM,SIZ,MSK
0000           SIZ...=1
0000           .IIF  NB,SIZ, SIZ...=SIZ
0000           .IF   NB,SYM
0000           MOD'SEP'V'SYM==BIT
0000           .IIF  NB,SIZ, MOD'SEP'S'SYM==SIZ
0000           .IIF  NB,MSK, MOD'SEP'M'SYM==<<<1@SIZ...>-1>@BIT...>
0000           .ENDC
0000           BIT...=BIT...+SIZ...
0000           .ENDM $VIELD1
0000           .IFF  DIF <LOCAL> <LOCAL>,ERROR ;ARG MUST BE "GLOBAL","LOCAL",OR NULL
0000 :
```

0000 .MACRO \$DEF SYM,ALLOC,SIZ
0000 .IIF NB,SYM,SYM:
0000 .IIF NB,ALLOC, ALLOC SIZ
0000 .ENDM SDÉF
0000 .MACRO \$SEQU SYM,VAL
SYM=VAL
0000 .ENDM \$SEQU
0000 .MACRO \$VIELD1 MOD,SEP,SYM,SIZ,MSK
SIZ...=1
0000 .IIF NB,SIZ, SIZ...=SIZ
0000 .IF NB,SYM
MOD'SEP'V 'SYM=BIT..
0000 .IIF NB,SIZ, MOD'SEP'S 'SYM=SIZ
0000 .IIF NB,MSK, MOD'SEP'M_ 'SYM=<<<1@SIZ...>-1>@BIT...>
.ENDC
BIT...=BIT...+SIZ...
.ENDM \$VIELD1
.ENDC

00000000 0000 .=0

00000000 0000 \$SEQU SHRS_FACILITY 0
SHRS_FACILITY=0

00001000 0000 \$SEQU SHRS_APPENDEDDB 4096
SHRS_APPENDEDDB=4096

00001008 0000 \$SEQU SHRS_APPENDEDDR 4104
SHRS_APPENDEDDR=4104

00001010 0000 \$SEQU SHRS_BADBYTE 4112
SHRS_BADBYTE=4112

00001018 0000 \$SEQU SHRS_BADFIELD 4120
SHRS_BADFIELD=4120

00001020 0000 \$SEQU SHRS_BADLONG 4128
SHRS_BADLONG=4128

00001028 0000 \$SEQU SHRS_BADWORD 4136
SHRS_BADWORD=4136

00001030 0000 \$SEQU SHRS_BEGIN 4144
SHRS_BEGIN=4144

00001038 0000 \$SEQU SHRS_BEGIND 4152
SHRS_BEGIND=4152

00001040 0000 \$SEQU SHRS_BEGINT 4160
SHRS_BEGINT=4160

00001048 0000 \$SEQU SHRS_CLICB 4168
SHRS_CLICB=4168

00001050 0000 \$SEQU SHRS_CLOSEIN 4176
SHRS_CLOSEIN=4176

00001058	0000	SEQU	SHRS_CLOSEOUT 4184
	0000		SHRS_CLOSEOUT=4184
00001060	0000	SEQU	SHRS_COPIEDB 4192
	0000		SHRS_COPIEDB=4192
00001068	0000	SEQU	SHRS_COPIEDR 4200
	0000		SHRS_COPIEDR=4200
00001070	0000	SEQU	SHRS_CREATED 4208
	0000		SHRS_CREATED=4208
00001078	0000	SEQU	SHRS_ENDED 4216
	0000		SHRS_ENDED=4216
00001080	0000	SEQU	SHRS_ENDEDD 4224
	0000		SHRS_ENDEDD=4224
00001088	0000	SEQU	SHRS_ENDEDT 4232
	0000		SHRS_ENDEDT=4232
00001090	0000	SEQU	SHRS_NEWFILES 4240
	0000		SHRS_NEWFILES=4240
00001098	0000	SEQU	SHRS_OPENIN 4248
	0000		SHRS_OPENIN=4248
000010A0	0000	SEQU	SHRS_OPENOUT 4256
	0000		SHRS_OPENOUT=4256
000010A8	0000	SEQU	SHRS_OVERLAY 4264
	0000		SHRS_OVERLAY=4264
000010B0	0000	SEQU	SHRS_READERR 4272
	0000		SHRS_READERR=4272
000010B8	0000	SEQU	SHRS_REPLACE 4280
	0000		SHRS_REPLACE=4280
000010C0	0000	SEQU	SHRS_WILDCONCAT 4288
	0000		SHRS_WILDCONCAT=4288
000010C8	0000	SEQU	SHRS_WILDOUTVER 4296
	0000		SHRS_WILDOUTVER=4296
000010D0	0000	SEQU	SHRS_WRITEERR 4304
	0000		SHRS_WRITEERR=4304
000010D8	0000	SEQU	SHRS_ABEND 4312
	0000		SHRS_ABEND=4312
000010E0	0000	SEQU	SHRS_ABENDD 4320
	0000		SHRS_ABENDD=4320
000010E8	0000	SEQU	SHRS_ABENDT 4328
	0000		SHRS_ABENDT=4328

000010F0	0000	SEQU	SHRS_SYSERRORPC 4336 SHRS_SYSERRORPC=4336
000010F8	0000	SEQU	SHRS_SYNTAX 4344 SHRS_SYNTAX=4344
00001100	0000	SEQU	SHRS_NOVALUE 4352 SHRS_NOVALUE=4352
00001108	0000	SEQU	SHRS_BADKEY 4360 SHRS_BADKEY=4360
00001110	0000	SEQU	SHRS_BADVALUE 4368 SHRS_BADVALUE=4368
00001118	0000	SEQU	SHRS_BADDELIM 4376 SHRS_BADDELIM=4376
00001120	0000	SEQU	SHRS_BADLOGIC 4384 SHRS_BADLOGIC=4384
00001128	0000	SEQU	SHRS_NOWILD 4392 SHRS_NOWILD=4392
00001130	0000	SEQU	SHRS_TEXT 4400 SHRS_TEXT=4400
00001138	0000	SEQU	SHRS_IDXCONCAT 4408 SHRS_IDXCONCAT=4408
00001140	0000	SEQU	SHRS_RELCONCAT 4416 SHRS_RELCONCAT=4416
00001148	0000	SEQU	SHRS_HIGHVER 4424 SHRS_HIGHVER=4424
00001150	0000	SEQU	SHRS_BADLOGICPC 4432 SHRS_BADLOGICPC=4432
00001158	0000	SEQU	SHRS_ATPC 4440 SHRS_ATPC=4440
00001160	0000	SEQU	SHRS_BADCOPIES 4448 SHRS_BADCOPIES=4448
00001168	0000	SEQU	SHRS_BADFORM 4456 SHRS_BADFORM=4456
00001170	0000	SEQU	SHRS_BADJOBID 4464 SHRS_BADJOBID=4464
00001178	0000	SEQU	SHRS_BADJOBNAME 4472 SHRS_BADJOBNAME=4472
00001180	0000	SEQU	SHRS_BADPRTY 4480 SHRS_BADPRTY=4480

00001188	0000	SEQU	SHR\$_BADQNAME 4488
	0000		SHR\$_BADQNAME=4488
00001190	0000	SEQU	SHR\$_BADTIME 4496
	0000		SHR\$_BADTIME=4496
00001198	0000	SEQU	SHR\$_NOQUEUE 4504
	0000		SHR\$_NOQUEUE=4504
000011A0	0000	SEQU	SHR\$_NOJOBID 4512
	0000		SHR\$_NOJOBID=4512
000011A8	0000	SEQU	SHR\$_NOJOBNAME 4520
	0000		SHR\$_NOJOBNAME=4520
000011B0	0000	SEQU	SHR\$_SYSERROR 4528
	0000		SHR\$_SYSERROR=4528
000011B8	0000	SEQU	SHR\$_NOTCOPIED 4536
	0000		SHR\$_NOTCOPIED=4536
000011C0	0000	SEQU	SHR\$_NOTCMPLT 4544
	0000		SHR\$_NOTCMPLT=4544
000011C8	0000	SEQU	SHR\$_RMSERROR 4552
	0000		SHR\$_RMSERROR=4552
000011D0	0000	SEQU	SHR\$_UNXPCTSTS 4560
	0000		SHR\$_UNXPCTSTS=4560
000011D8	0000	SEQU	SHR\$_HASHCONCAT 4568
	0000		SHR\$_HASHCONCAT=4568
000011E0	0000	SEQU	SHR\$_INCOMPAT 4576
	0000		SHR\$_INCOMPAT=4576
000011E8	0000	SEQU	SHR\$_VALERR 4584
	0000		SHR\$_VALERR=4584
000011F0	0000	SEQU	SHR\$_FILNOTDEL 4592
	0000		SHR\$_FILNOTDEL=4592
000011F8	0000	SEQU	SHR\$_CONFDEL 4600
	0000		SHR\$_CONFDEL=4600
00001200	0000	SEQU	SHR\$_DELETED 4608
	0000		SHR\$_DELETED=4608
00001208	0000	SEQU	SHR\$_DELVER 4616
	0000		SHR\$_DELVER=4616
00001210	0000	SEQU	SHR\$_PURGEVER 4624
	0000		SHR\$_PURGEVER=4624
00001218	0000	SEQU	SHR\$_CLOSEDEL 4632
	0000		SHR\$_CLOSEDEL=4632

00001220	0000	\$SEQU	SHRS_DIRTOOBUS 4640
	0000		SHRS_DIRTOOBUS=4640
00001228	0000	\$SEQU	SHRS_NOFILPURG 4648
	0000		SHRS_NOFILPURG=4648
00001230	0000	\$SEQU	SHRS_FILNOTPUR 4656
	0000		SHRS_FILNOTPUR=4656
00001238	0000	\$SEQU	SHRS_SEARCHFAIL 4664
	0000		SHRS_SEARCHFAIL=4664
00001240	0000	\$SEQU	SHRS_DELINTERR 4672
	0000		SHRS_DELINTERR=4672
00001248	0000	\$SEQU	SHRS_PARSEFAIL 4680
	0000		SHRS_PARSEFAIL=4680
00001250	0000	\$SEQU	SHRS_FILPURGED 4688
	0000		SHRS_FILPURGED=4688
00001258	0000	\$SEQU	SHRS_ENDABORT 4696
	0000		SHRS_ENDABORT=4696
00001260	0000	\$SEQU	SHRS_ENDDIAGS 4704
	0000		SHRS_ENDDIAGS=4704
00001268	0000	\$SEQU	SHRS_ENDNOOBJ 4712
	0000		SHRS_ENDNOOBJ=4712
00001270	0000	\$SEQU	SHRS_HALTED 4720
	0000		SHRS_HALTED=4720
00001278	0000	\$SEQU	SHRS_NOCMDMEM 4728
	0000		SHRS_NOCMDMEM=4728
00001280	0000	\$SEQU	SHRS_QEMPTY 4736
	0000		SHRS_QEMPTY=4736
00001288	0000	\$SEQU	SHRS_CBT 4744
	0000		SHRS_CBT=4744
00001290	0000	\$SEQU	SHRS_EXISTS 4752
	0000		SHRS_EXISTS=4752
00001298	0000	\$SEQU	SHRS_UNLOCKED 4760
	0000		SHRS_UNLOCKED=4760
000012A0	0000	\$SEQU	SHRS_RENAMED 4768
	0000		SHRS_RENAMED=4768
000012A8	0000	\$SEQU	SHRS_PROTECTED 4776
	0000		SHRS_PROTECTED=4776
000012B0	0000	\$SEQU	SHRS_NOTLOCKED 4784
	0000		SHRS_NOTLOCKED=4784

000012B8	0000	SEQU	SHRS_ACTIMAGE 4792 SHRS_ACTIMAGE=4792
000012C0	0000	SEQU	SHRS_DIRNOTCRE 4800 SHRS_DIRNOTCRE=4800
000012C8	0000	SEQU	SHRS_NODESTQUE 4808 SHRS_NODESTQUE=4808
000012D0	0000	SEQU	SHRS_ILLDESQUE 4816 SHRS_ILLDESQUE=4816
000012D8	0000	SEQU	SHRS_NOTTERM 4824 SHRS_NOTTERM=4824
000012E0	0000	SEQU	SHRS_CONFQUAL 4832 SHRS_CONFQUAL=4832
000012E8	0000	SEQU	SHRS_ILLDIRCOPY 4840 SHRS_ILLDIRCOPY=4840
000012F0	0000	SEQU	SHRS_INSVIRMEM 4848 SHRS_INSVIRMEM=4848
000012F8	0000	SEQU	SHRS_CREATEDSTM 4856 SHRS_CREATEDSTM=4856
00001300	0000	SEQU	SHRS_NOTRUNC 4864 SHRS_NOTRUNC=4864
00001308	0000	SEQU	SHRS_PRODNOTINS 4872 SHRS_PRODNOTINS=4872
00001310	0000	SEQU	SHRS_TOTAL 4880 SHRS_TOTAL=4880
00001318	0000	SEQU	SHRS_FILPURG 4888 SHRS_FILPURG=4888
00001320	0000	SEQU	SHRS_FILDEL 4896 SHRS_FILDEL=4896
00001328	0000	SEQU	SHRS_INVQUAVAL 4904 SHRS_INVQUAVAL=4904
00001330	0000	SEQU	SHRS_NOFILES 4912 SHRS_NOFILES=4912
00001338	0000	SEQU	SHRS_FILNOTACC 4920 SHRS_FILNOTACC=4920
00001340	0000	SEQU	SHRS_QUALMISS 4928 SHRS_QUALMISS=4928
00001348	0000	SEQU	SHRS_FILSPCSRCH 4936 SHRS_FILSPCSRCH=4936

00001350 0000 \$SEQU SHRS_NOSRCHLST 4944
0000 0000 SHRS_NOSRCHLST=4944

00001358 0000 \$SEQU SHRS_NOSUCHID 4952
0000 0000 SHRS_NOSUCHID=4952

00000000 0000 \$DEFEND SHR,,DEF
.MACRO SSHRDEF A
.ENDM SSHRDEF
.IIF DIF <> <GLOBAL>,..DISABLE SUPPRESSION
.CROSS
.RESTORE

00000000 0000 .ENDC
\$\$GBL = 0
.IIF IDN,LOCAL,GLOBAL,\$\$GBL = 1
.IRP MSGPAIR, <<ATPC,INFO>>
\$SHR_MSGCOD REM, REMS_FACILITY, MSGPAIR
.ENDR
\$SHR_MSGCOD REM, REMS_FACILITY, ATPC,INFO
.IF IDN,INFO,WARNING
.IF EQ \$\$GBL
REMS_ATPC = 0
.IFF
REMS_ATPC == 0
.ENDC
.IFF
.IF IDN,INFO,SUCCESS
.IF EQ \$\$GBL
REMS_ATPC = 1
.IFF
REMS_ATPC == 1
.ENDC
.IFF
.IF IDN,INFO,ERROR
.IF EQ \$\$GBL
REMS_ATPC = 2
.IFF
REMS_ATPC == 2
.ENDC
.IFF
.IF IDN,INFO,INFO
.IF EQ \$\$GBL
REMS_ATPC = 3
.IFF
REMS_ATPC == 3
.ENDC
.IFF
.IF IDN,INFO,SEVERE
.IF EQ \$\$GBL
REMS_ATPC = 4
.IFF
REMS_ATPC == 4
.ENDC
.IFF
.IF EQ \$\$GBL

0000 REM\$_ATPC = INFO
0000 .IFF
0000 REM\$_ATPC == INFO
0000 .ENDC
0000 .ENDC
0000 .ENDC
0000 .ENDC
0000 .ENDC
0000 .IF EQ \$\$GBL
0000 REM\$_ATPC = REM\$_ATPC+SHRS\$_ATPC+<REM\$_FACILITY@16>
0000 .IFF
0000 REM\$_ATPC == REM\$_ATPC+SHRS\$_ATPC+<REM\$_FACILITY@16>
0000 .ENDC

0000 141 .noshow meb
0000 142
0000 143 :
0000 144 ;

0000 146 .SBTTL RTPAD - MAIN ROUTINE
0000 147 ++
0000 148 FUNCTIONAL DESCRIPTION:
0000 149
0000 150 MAIN ROUTINE
0000 151
0000 152 CALLING SEQUENCE:
0000 153
0000 154 RUN FROM A TERMINAL
0000 155
0000 156 INPUT PARAMETERS:
0000 157
0000 158 NONE
0000 159
0000 160 IMPLICIT INPUTS:
0000 161
0000 162 NONE
0000 163
0000 164 OUTPUT PARAMETERS:
0000 165
0000 166 NONE
0000 167
0000 168 IMPLICIT OUTPUTS:
0000 169
0000 170 NONE
0000 171
0000 172 COMPLETION CODES:
0000 173 RETURNED IF A SYSTEM SERVICE HAS AN UNEXPECTED ERROR
0000 174
0000 175 SIDE EFFECTS:
0000 176
0000 177 DECNET LINK SET UP WITH A REMOTE NODE
0000 178
0000 179
0000 180 --
0000 181 RTPAD: .WORD 0
0002 182
0002 183
0002 184 .if df debug
0002 185 BSBW DEBUG_SETUP ; enable ^B
0002 186 .endc
0002 187
0002 188 ;
0002 189 ; INIT creates logical link and starts up protocol module.
0002 190 ; everything after that is AST driven.
0002 191 ;
0002 192
0000 30 0002 193 BSBW INIT ; INIT & set up logical link
0005 194
0005 195 10\$: \$HIBER_S ; Proceed asynchronous from now on
0005
0005 GLOBL SYSSHIBER
0005 CALLS #0,G^SYS\$HIBER
02B1'CF 95 000C 196 TSTB WAKEFLAG ; Time to exit if > 0
F3 13 0010 197 BEQL 10\$; Spurious \$WAKE
0012 198
11 02B2'CF E9 0012 199 BLBC CTERM_FLAG,20\$; branch if not cterm

```

0000'CF      D5 0017 200      TSTL    WRITEQIO          ; must also be zero before exiting
0B   13 001B 201      BEQL    20$                ; yup, exit
001D 001D 202      $SETAST_S          ENBFLG = #1      ; Allow ast delivery (turned off by QUIT)
001D
001F
0026
0026
0028 0028 203      BRB    10$                ; loop
0C28 204      20$:                                ; Wakeflag is set, exit back to DCL
0028 205
0028 206
0028 207
0028 208
0028 209      $SETAST_S          ENBFLG = #0      ; Shut down ast delivery
0028
0028
0031 0031 210      .GLOBL  SYSSSETAST
0031 211      PUSHL  CALLS  #0
0031
0031
0031 0031 211      $CANCEL_S          CHAN = READCHAN
0031
0031
0031 0031 211      .GLOBL  SYSSCANCEL
0031
0031
0031 0031 211      MOVZWL CALLS  READCHAN,-(SP)
0031
0031
0031 0031 211      .GLOBL  SYSSCANCEL
0031
0031
0031 0031 211      MOVZWL CALLS  #1,G^SYSSCANCEL
0031
0031
0031 0031 211      $CANCEL_S          CHAN = TERMMBXCHAN
0031
0031
0031 0031 211      .GLOBL  SYSSCANCEL
0031
0031
0031 0031 211      MOVZWL CALLS  TERMMBXCHAN,-(SP)
0031
0031
0031 0031 211      .GLOBL  SYSSCANCEL
0031
0031
0031 0031 211      MOVZWL CALLS  #1,G^SYSSCANCEL
0031
0031
0031 0031 211      $CANCEL_S          CHAN = MAILCHAN
0031
0031
0031 0031 211      .GLOBL  SYSSCANCEL
0031
0031
0031 0031 211      MOVZWL CALLS  MAILCHAN,-(SP)
0031
0031
0031 0031 211      .GLOBL  SYSSCANCEL
0031
0031
0031 0031 211      MOVZWL CALLS  #1,G^SYSSCANCEL
0031
0031
0031 0031 211      $PUTMSG_S MSGVEC = EXITMSG      ; Tell user why
0031
0031
0031 0031 211      .GLOBL  SYSSPUTMSG
0031
0031
0031 0031 211      PUSHL  #0
0031
0031
0031 0031 211      $PUSHADR 0,CONTEXT=Q
0031
0031
0031 0031 211      .IF    IDN,0,0
0031
0031
0031 0031 211      PUSHL  #0
0031
0031
0031 0031 211      .IFF
0031
0031
0031 0031 211      PUSHAQ  0
0031
0031
0031 0031 211      .ENDC
0031
0031
0031 0031 211      $PUSHADR 0
0031
0031
0031 0031 211      .IF    IDN,0,0
0031
0031
0031 0031 211      PUSHL  #0
0031
0031
0031 0031 211      .IFF
0031
0031
0031 0031 211      PUSHAL  0
0031
0031
0031 0031 211      .ENDC
0031
0031
0031 0031 211      .ENDC

```


					CLRQ -(SP)
					: IFF
					PUSHL #0
					\$PUSHADR 0
					.ENDC
					\$PUSHADR 0, CONTEXT=0
					: IF IDN,0,0
					PUSHL #0
					: IFF
					PUSHAQ 0
					.ENDC
7E	00A3'8F	3C	007E		MOVZWL #IOS_SETMODE!IOSM_CTRLYAST,-(SP)
7E	0154'CF	3C	0083		MOVZWL CNTRECHAN,-(SP)
00	DD	00	0088		PUSHL #0
00000000'GF	OC	FB	008A		CALLS #12,G^SYSSQIO
			0091		
			0091	220	
			0091	221	
			0091	222	: Restore original out of band and resource wait mode
			0091	223	
031C'CF	7E	D4	0091	224	CLRL -(SP)
00000000'GF	02	DF	0093	225	PUSHL OLDCTRL
		FB	0097	226	CALLS #2,G^LIB\$ENABLE_CTRL
			009E	227	; Reenable cli out of band characters
			009E	228	
			009E	229	\$SETRWM_S -
					WATFLG = OLDSETRWM
					: Set resource wait mode
					: to whatever it was upon entry
0320'CF	00000000'GF	DD	009E		
00000000'GF	01	FB	00A2		
			00A9		
			FF54'	30	
				00A9	00A9
				00AC	230
				00AC	231
				00AC	232
50	0309'CF	D0	00AC	233	BSBW CTERM\$CLOSE_LOG
50	00000000'8F	D1	00B1	234	; Close log file if open
1A	12	00B8	235	CMPL #\$\$\$_EXQUOTA, R0	
		00BA	236	BNEQ 100\$	
			030D'CF	DD	RETSTATUS, R0
			01	DD	; Get saved status
			01FE115B	8F	#SSS_EXQUOTA, R0
			50	DD	; Exceeded some quota?
00000000'GF	04	FB	00C0	237	BNEQ nope, exit
			00C6	238	
			00C8	239	PUSHL QUIT_PC
			00CF	240	PUSHL #1
			00D4	241	PUSHL #REMS_ATPC
			00D5	242	PUSHL R0
50	0309'CF	D0	00CF	243	CALLS #4,G^LIB\$SIGNAL
50	0309'CF	D0	00D4	244	; signal error
			00D5	245	MOVL RETSTATUS, R0
				100\$:	RET
					; Get saved status
					; Exit program

00D5 247 .SBTTL INIT - INITIALIZATION OF LINK, ETC.
00D5 248 ++
00D5 249 FUNCTIONAL DESCRIPTION:
00D5 250 PERFORMS INITIALIZATION FUNCTIONS FOR RTPAD
00D5 251
00D5 252
00D5 253
00D5 254
00D5 255
00D5 256
00D5 257
00D5 258
00D5 259
00D5 260
00D5 261
00D5 262
00D5 263
00D5 264
00D5 265
00D5 266
00D5 267
00D5 268
00D5 269
00D5 270
00D5 271 CHANNEL NUMBERS, ETC.
00D5 272
00D5 273
00D5 274
00D5 275 WILL RETURN COMPLETION CODES OF SYSTEM SERVICES WITH UNEXPECTED ERRORS
00D5 276
00D5 277
00D5 278
00D5 279
00D5 280
00D5 281 :--
00D5 282
00D5 283 INIT:
00D5 284 \$TRNLOG_S - : Translate a logical name
00D5 285 LOGNAME = SYSSNODE, - : from SYSSNODE
00D5 286 RSLBUF = NODENAME, - : to the ending message
00D5 287 RSLLEN = NODENAME : setting the correct length
00D5 288 .GLOBL SYS\$TRNLOG
00D5 289 PUSHL #0
00D5 290 \$PUSHADR 0,CONTEXT=B
00D5 291 .IF IDN.0.0
00D5 292 PUSHL #0
00D5 293 .IFF
00D5 294 PUSHAB 0
00D5 295 .ENDC
00D5 296
00D5 297 \$PUSHADR 0,CONTEXT=B
00D5 298 .IF IDN.0.0
00D5 299 PUSHL #0
00D5 300 .IFF
00D5 301 PUSHAB 0
00D5 302 .ENDC
00 DD 00D5
00 DD 00D7
00 DD 00D7
00 DD 00D9
00 DD 00D9
00 DD 00D9
00 DD 00D9
00 DD 00DB
00 DD 00DB
00 DD 00DB
00 DD 00DB

```

00DB      $PUSHADR NODENAME,CONTEXT=Q
00DB      :IF IDN,0,NODENAME
00DB      PUSHL #0
00DB      :IFF
00DB      PUSHAQ NODENAME
00DF      .ENDC

01B0'CF  7F  00DB      $PUSHADR NODENAME,CONTEXT=W
00DF      :IF IDN,0,NODENAME
00DF      PUSHL #0
00DF      :IFF
00DF      PUSHAW NODENAME
00DF      .ENDC

01B0'CF  3F  00E3      $PUSHADR SYSSNODE,CONTEXT=Q
00E3      :IF IDN,0,SYSSNODE
00E3      PUSHL #0
00E3      :IFF
00E3      PUSHAQ SYSSNODE
00E3      .ENDC

04C6'CF  7F  00E3      CALLS #6,G^SYS$TRNLOG
00E7
00E7

00000000'GF  06  FB  00E7      ONERROR RET          ; Exit on error
00EE      288      BLBS R0,30000$ 
01 50  E8  00EE      RET
04 00F1      288      30000$:
00F2      289      STRNLOG_S -           ; translate 'RTPADSLOG'
00F2      290      LOGNAM = RTPAD_LOGNAM,-
00F2      291      RSLLEN = RTLOG_DESC,-
00F2      292      RSLBUF = RTLOG_DESC
00F2      293      .GLOBL SYS$TRNLOG
00F2      PUSHL #0
00F4      $PUSHADR 0,CONTEXT=B
00F4      :IF IDN,0,0
00F4      PUSHL #0
00F6      :IFF
00F6      PUSHAB 0
00F6      .ENDC

00  DD  00F6      $PUSHADR 0,CONTEXT=B
00  DD  00F6      :IF IDN,0,0
00  DD  00F6      PUSHL #0
00  DD  00F8      :IFF
00  DD  00F8      PUSHAB 0
00  DD  00F8      .ENDC

00FB'CF  7F  00F8      $PUSHADR RTLOG_DESC,CONTEXT=Q
00FB'CF  00F8      :IF IDN,0,RTLOG_DESC
00FB'CF  00F8      PUSHL #0
00FB'CF  00F8      :IFF
00FB'CF  00F8      PUSHAQ RTLOG_DESC
00FB'CF  00F8      .ENDC

02BA'CF  7F  00FC      $PUSHADR RTLOG_DESC,CONTEXT=W

```

```

02BA'CF 3F 00FC      IF IDN,0,RTLOG_DESC
                  PUSHL #0
                  IFF
                  PUSHAW RTLOG_DESC
                  .ENDC

                  SPUSHADR RTPAD_LOGNAM,CONTEXT=Q
                  :IF IDN,0,RTPAD_LOGNAM
                  PUSHL #0
                  IFF
02D2'CF 7F 0100      PUSHAQ RTPAD_LOGNAM
                  .ENDC

00000000'GF 06 FB 0104      CALLS #6,G^SYS$TRNLOG

0000'8F 50 B1 010B 294      CMPW R0,#$$$NOTRAN
1E 13 0110 295      BEQL $S
1B 50 E9 0112 296      BLBC R0,$S
                  : continue if no definition
                  : or error

                  0115 297      : translate hex byte string to binary value
                  0115 298
                  0115 299

02B6'CF DF 0115 300      PUSHAL RTLOG_FLAGS
02C2'CF 9F 0119 301      PUSHAB RTLOG_BUF
                  : flags
                  : string
7E 02BA'CF 3C 011D 302      MOVZWL RTLOG_DESC,-(SP)
                  : length
00000000'GF 03 FB 0122 303      CALLS #3,G^IB$CVT_HTB
                  : convert hex to binary
04 50 E8 0129 304      BLBS R0,$S
                  : go if ok
02B6'CF D4 012C 305      CLRL RTLOG_FLAGS
                  : otherwise zero

                  0130 306
                  0130 307 $S:      $GETDVI_S -
                  0130 308          : Get the device characteristics
                  0130 309          : of the translated SY$INPUT
                  :DEVNAM = TTYDESC, -
                  :ITMLST = DVILIST
                  :GLOBL SY$GETDVI
                  :SASNPUT 0,#0
                  :$ST1 = 0
                  :IF IDN,<0>,<0>
                  :IF IDN,<#0>,<#0>
                  :$ST1 = 1
                  :ENDC
                  :ENDC
                  :IF NE $ST1
                  CLRQ -(SP)
                  IFF
                  SPUSHADR 0,CONTEXT=Q
                  PUSHL #0
                  .ENDC

                  SPUSHADR 0
                  :IF IDN,0,0
                  PUSHL #0
                  IFF
                  PUSHAL 0
                  .ENDC

                  SPUSHADR 0,CONTEXT=Q
                  :IF IDN,0,0
                  PUSHL #0
                  .IFF

```



```

0158          .ENDC
0158
0158          $PUSHADR TTYDESC,CONTEXT=Q
0158          .IF IDN,0,TTYDESC
0158          PUSHL #0
0158          .IFF
0158          PUSHAQ TTYDESC
0158          .ENDC
04D6'CF    7F 0158
015C
015C
00000000'GF  05  FB 015C
0163          CALLS #5,G^SYSS$GETDEV
01 50  E8 0163  315  ONERROR RET
04 0163
0166          BLBS R0,30002$
0167          RET
0167          30002$:
0060'CF    0054'CF 90 0167  316  MOVB DEVCLASS_TEMP,DEVCLASS ; Pack the data correctly
0061'CF    0058'CF 90 016E  317  MOVB DEVTYPE_TEMP,DEVTYPE
0062'CF    005C'CF B0 0175  318  MOVW DEVBUSIZ_TEMP,DEVBUSIZ
017C
00'8F     0060'CF 91 017C  321  CMPB DEVCLASS, #DCS_TERM ; Is it a terminal?
12   13 0182  322  BEQL 10$ ; Yes
0184          $PUTMSG_S - ; Output an error message
0184          MSGVEC = NOTTERM ; saying SYSS$COMMAND not a terminal
0184          GLOBL SYSS$PUTMSG
0184          PUSHL #0
0186          $PUSHADR 0,CONTEXT=Q
0186          .IF IDN,0,0
0186          PUSHL #0
0188          .IFF
0188          PUSHAQ 0
0188          .ENDC
0188
00   DD 0184
0186
0186          $PUSHADR 0
0186          .IF IDN,0,0
0186          PUSHL #0
018A          .IFF
018A          PUSHAL 0
018A          .ENDC
018A
018A          $PUSHADR NOTTERM
018A          .IF IDN,0,NOTTERM
018A          PUSHL #0
018A          .IFF
011C'CF    DF 018A          PUSHAL NOTTERM
018E
018E
00000000'GF  04  FB 018E
0195          CALLS #4,G^SYSS$PUTMSG
04 0195  325  RET
0196
0196
0196  326
0196  327
0196  328 ; Call RTL routine to assign channel and associate mailbox to terminal
0196  329 ;
0196  330 10$: PUSHAW TERMMBXCHAN ; Arg #5 is the terminal mailbox chan
0158'CF    3F 0196  331

```

```

014C'CF 3F 019A 332 PUSHAW READCHAN : Arg #4 is the terminal input channel
0764'CF DF 019E 333 PUSHAL MAXMSGSZ : Arg #3 is the message buffer quota
0764'CF DF 01A2 334 PUSHAL MAXMSGSZ : Arg #2 is the maximum message size
04D6'CF 7F 01A6 335 PUSHAQ TTYDESC : Arg #1 is the terminal device name
00000000'GF 05 FB 01AA 336 CALLS #5, G^LIB$ASN_WTH_MBX ; Assign a channel w/ a mailbox
01 50 E8 01B1 337 ONERROR RET ; Die if any error
04 01B4
01B5
01B5 338
01B5 339 ; Assign a terminal write channel
01B5 340
01B5 341 ;$ASSIGN_S -
01B5 342 DEVNAM = TTYDESC, - ; Assign a channel
01B5 343 CHAN = WRITECHAN ; to the terminal device
01B5 .GLOBL SYSS$ASSIGN ; for terminal output
01B5 $ASN$PUSH 0,#0
00000000 01B5 $$T1 = 0
01B5 .IF IDN,<0>,<0>
00000001 01B5 .IF IDN,<#0>,<#0>
01B5 $$T1 = 1
01B5 .ENDC
01B5 .ENDC
00000001 01B5 .IF NE $$T1
7E 7C 01B5 CLRQ -(SP)
01B7
01B7 $PUSHADR 0,CONTEXT=Q
01B7 PUSHL #0
01B7 .ENDC
01B7
01B7 $PUSHADR WRITECHAN,CONTEXT=W
01B7 .IF IDN,0,WRITECHAN
01B7 PUSHL #0
01B7 .IFF
0150'CF 3F 01B7 PUSHAW WRITECHAN
01B7 .ENDC
01BB
01BB $PUSHADR TTYDESC,CONTEXT=Q
01BB .IF IDN,0,TTYDESC
01BB PUSHL #0
01BB .IFF
04D6'CF 7F 01BB PUSHAQ TTYDESC
01BF .ENDC
01BF
00000000'GF 04 FB 01BF CALLS #4,G^SYSS$ASSIGN
01C6
01C6 344 ONERROR RET ; Die if any error
01 50 E8 01C6 BLBS R0,30004$ ; Die if any error
04 01C9
01CA
01CA 345 ; See if SYSS$INPUT is a file
01CA 346
01CA 347
01CA 348 ;OPEN FAB = SYSINFAB ; Open SYSS$INPUT
01CA $RMSCALL OPEN,SYSINFAB,, ; Open SYSS$INPUT

```



```

01F8      .IF      NB <>
01F8      $$TMP=1
01F8      .ENDC
01F8      .IF      NE $$TMP
01F8      .ERROR
01F8      .ENDC
01F8      .ENDC
01F8      .IF      NB <SYSINFAB>
01F8      $$TMP1=1
01F8      .IF      NB <>
01F8      PUSHAL
01F8      $$TMP1=3
01F8      .ENDC
01F8      .IF      NB <>
01F8      PUSHAL
01F8      .IF      EQ <$$TMP1-1>
01F8      $$TMP1=2
01F8      .ENDC
01F8      .IFF
01F8      .IF      EQ <$$TMP1-3>
01F8      PUSHL #0
01F8      .ENDC
01F8      .ENDC
01F8      .NTYPE  $$TMP2,SYSINFAB
01F8      .IF      EQ <<$$TMP2&^XF0>-^X50>
01F8      PUSHL SYSINFAB
01F8      .IFF
01F8      .IF      EQ <<$$TMP2&^XF0>-^X10>
01F8      PUSHL SYSINFAB
01F8      .IFF
01F8      PUSHAL SYSINFAB
01F8      .ENDC
01F8      .ENDC
01F8      CALLS  #$$TMP1,G^SYSSCLOSE
01F8      .ENDC

0203      358 30$:
0203      359
0203      360      : Check for /LOG [=filespec]
0203      361
0203      362
0203      363      PUSHAQ LOG_DESC
0203      364      CALLS #1,G^CLI$PRESENT      ; LOG label
0203      365      BLBC R0,62$      ; See if present
0203      366      PUSHAQ LOG_FILE_DESC      ; branch if not present
0203      367      PUSHAQ LOG_DESC-
0203      368      CALLS #2,G^CLI$GET_VALUE      ; return buffer
0203      369      BLBC R0,61$      ; LOG label
0203      370      PUSHAQ LOG_FILE_DESC      ; get value
0203      371      CALLS #1,G^TERMS$OPEN_LOG      ; continue if no error
0203      372      BLBS R0,62$      ; use this file
0203      373      RET      ; Open log file
0203      374      61$:      ; Branch if ok
0203      375      62$:      ; exit on error
0203      376
0203      377

```

```

0198'CF 0232 378 ; Get node name from CLI
017D'CF 0232 379
00000000'GF 02 FB 0236 380 PUSHAQ NODE_NAME_DESC
01 50 E8 0241 381 PUSHAQ NODEDESC
04 0244 382 CALLS #2, G^CLISGET_VALUE
0253 383 ONERROR REF ; Parameter name
0253 384 BLBS R0,30007$ ; Get node name
0253 385 RET ; Exit on error
0253 386 30007$:
0253 387
0253 388
0172'CF 0245 389 PUSHAQ DTE_DESC
00000000'GF 01 FB 0249 390 CALLS #1, G^CLISPRESSENT ; DTE label
29 50 E9 0250 391 BLBC R0, NOT_DTE ; See if present
0253 392
0253 393
0253 394 ; branch if not present
0253 395
50 01B0'CF 3C 0253 396 MOVZWL NODENAME,R0 ; length
50 01B4'CF C0 0258 397 ADDL2 NODENAME+4,R0 ; plus address
01B0'CF 02 A0 025D 398 ADDW #2,NODENAME ; add to length
60 ODOA 8F B0 0262 399 MOVW #^XODOA,(R0) ; CR,LF
0267 400
7E 0150'CF 3C 0267 401 MOVZWL WRITECHAN,-(SP) ; command channel
0198'CF 026C 402 PUSHAQ NODE NAME DESC ; value of P1
00000000'GF 02 FB 0270 403 CALLS #2, G^TERM$EMULATE ; call terminal emulation code
0277 404 ONERROR RET ; exit immediately on error
01 50 E8 0277 405 BLBS R0,30008$ ; otherwise, return to hiber code
04 027A 406
027B 407 NOT_DTE: RSB ; Assign a channel for ^C and ^Y handling
027C 408
027C 409
027C 410
027C 411 $ASSIGN_S - ; Assign a channel
027C 412 DEVNAM = TTYDESC, - ; to the terminal device
027C 413 CHAN = CNTRLCHAN ; for control (AST's)
027C 414 GLOBL SYSS$ASSIGN
027C 415 $ASN$PUSH 0,#0
027C 416 $ST1 = 0
027C 417 .IF IDN,<0>,<0>
027C 418 .IF IDN,<#0>,<#0>
027C 419 $ST1 = 1
027C 420 .ENDC
027C 421 .ENDC
027C 422 .IF NE $ST1
027C 423 CLRQ -(SP)
027C 424 .IFF
027C 425 $PUSHADR 0,CONTEXT=Q
027E 426 PUSHL #0

```

```

027E .ENDC
027E
027E $PUSHADR CNTRLCHAN,CONTEXT=W
027E .IF IDN,0,CNTRLCHAN
027E PUSHL #0
027E .IFF
027E PUSHAW CNTRLCHAN
027E .ENDC

0282
0282 $PUSHADR TTYDESC,CONTEXT=Q
0282 .IF IDN,0,TTYDESC
0282 PUSHL #0
0282 .IFF
0282 PUSHAQ TTYDESC
0282 .ENDC

00000000'GF 04 FB 0286 CALLS #4,G^SYSS$ASSIGN
00000000'GF 04 FB 028D
01 50 E8 028D 414 ONERROR RET ; Die if any error
01 50 E8 028D BLBS R0,30009$ ; 
01 50 E8 0290 RET
01 50 E8 0291 30009$:

0291 415
0291 416
0291 417 : Loop through node name to remove trailing :
0291 418
0291 419
52 0198'CF 7D 0291 420 MOVQ NODE_NAME_DESC, R2 ; Get node name length, address
52 52 3C 0296 421 MOVZWL R2 R2 and isolate its real length
0A 13 0299 422 BEQL 50$ ; No length??
3A FF A342 91 029B 423 40$: CMPB -1(R3)[R2], #^A':'
09 12 02A0 424 BNEQ 60$ ; A trailing colon?
F6 52 F5 02A2 425 SOBGTR R2, 40$ ; Nope
50 0000'8F 3C 02A5 426 50$: MOVZWL #SS$_NOSUCHNODE, R0 ; Yep, remove it from count and loop
04 02AA 427 RET ; Error
0000'CF 52 7D 02AB 428 ; and exit
02B0 429 60$: MOVQ R2, REMOTENODE ; Save the node name descriptor
02B0 430
02B0 431
02B0 432 : Form the network connection string
02B0 433
02B0 434
0189'CF 7F 02B0 435 PUSHAQ OBJ_DESC ; Arg #3 is the right part (obj type)
0000'CF 7F 02B4 436 PUSHAQ REMOTENODE ; Arg #2 is the left part (node name)
01A8'CF 7F 02B8 437 PUSHAQ CONNDESC ; Arg #1 is the resultant string
00000000'GF 03 FB 02BC 438 CALLS #3, G^STR$CONCAT ; Go concatenate for connection string
01 50 E8 02C3 439 ONERROR RET ; Exit on error
01 50 E8 02C3 BLBS R0,30010$ ; 
01 50 E8 02C6 RET
01 50 E8 02C7 30010$:
02C7
02C7
02C7 ADDL3 R2, CONNDESC+4, R8 ; Address just beyond node name
58 01AC'CF 52 C1 02C7 440 ADDL S^#OBJ_C_PREFIX, R8 ; then offset to object number
58 03 CO 02CD 441
02D0 442
02D0 443 :
02D0 444 : *** TEMPORARY CODE TO DETECT /OLD QUALIFIER

```

```

015C'CF 02D0 445 : pushaq old_desc      ; *** TEMP
00000000'GF 01 FB 02D4 446 calls #1, g^cli$present ; *** TEMP
          05 50 E9 02DB 447 blbc r0, 65$    ; *** TEMP; /NOOLD, go try 42 first
68 3332 8F B0 02DE 448 movw "#A'23', (r8) ; *** TEMP; /OLD, change to 23
          02E3 449
          02E3 450 65$: pushaq old_desc      ; *** TEMP
          02E3 451
          02E3 452
          02E3 453
          02E3 454
          02E3 455 : Connect to the remote node by requesting a logical link
          02E3 456 70$: PUSHAQ PSTRU_MSG      ; Address the PSTRU message desc
00000000'GF 01 FB 02E7 457 CALLS #1, G^STR$FREE1_DX ; and free up anything in it
          02EE 458 ONERROR RET    ; Exit on error
          01 50 E8 02EE 459
          04 02F1 460
          02F2 461
          02F2 462
          0764'CF 3F 02F2 463
          0421'CF 9F 02F6 464
          0200'CF 7F 02FA 465
          01F8'CF 7F 02FE 466
          0148'CF 3F 0302 467
          0144'CF 3F 0306 468
          01A8'CF 7F 030A 469
00000000'GF 07 FB 030E 470
13 50 E8 0315 471
          0318 472
          0318 473
          0318 474
          0318 475
          0318 476
          0318 477
          0318 478
          0318 479
          0318 480 80$: BSBW 170$      ; If object type 42 failed then try object type 23
          0318 481
          0318 482
          0318 483
          0318 484
          0318 485
          0318 486
          0318 487
          0318 488
          0318 489
          0318 490
          0318 491
          0318 492
          0318 493
          032E 494
          032E 495
          032E 496
          032E 497
          032E 498
          032E 499
          032E 500
          032E 501
          032E 502
          032E 503
          032E 504
          032E 505
          032E 506
          032E 507
          032E 508
          032E 509
          032E 510
          032E 511
          032E 512
          032E 513
          032E 514
          032E 515
          032E 516
          032E 517
          032E 518
          032E 519
          032E 520
          032E 521
          032E 522
          032E 523
          032E 524
          032E 525
          032E 526
          032E 527
          032E 528
          032E 529
          032E 530
          032E 531
          032E 532
          032E 533
          032E 534
          032E 535
          032E 536
          032E 537
          032E 538
          032E 539
          032E 540
          032E 541
          032E 542
          032E 543
          032E 544
          032E 545
          032E 546
          032E 547
          032E 548
          032E 549
          032E 550
          032E 551
          032E 552
          032E 553
          032E 554
          032E 555
          032E 556
          032E 557
          032E 558
          032E 559
          032E 560
          032E 561
          032E 562
          032E 563
          032E 564
          032E 565
          032E 566
          032E 567
          032E 568
          032E 569
          032E 570
          032E 571
          032E 572
          032E 573
          032E 574
          032E 575
          032E 576
          032E 577
          032E 578
          032E 579
          032E 580
          032E 581
          032E 582
          032E 583
          032E 584
          032E 585
          032E 586
          032E 587
          032E 588
          032E 589
          032E 590
          032E 591
          032E 592
          032E 593
          032E 594
          032E 595
          032E 596
          032E 597
          032E 598
          032E 599
          032E 600
          032E 601
          032E 602
          032E 603
          032E 604
          032E 605
          032E 606
          032E 607
          032E 608
          032E 609
          032E 610
          032E 611
          032E 612
          032E 613
          032E 614
          032E 615
          032E 616
          032E 617
          032E 618
          032E 619
          032E 620
          032E 621
          032E 622
          032E 623
          032E 624
          032E 625
          032E 626
          032E 627
          032E 628
          032E 629
          032E 630
          032E 631
          032E 632
          032E 633
          032E 634
          032E 635
          032E 636
          032E 637
          032E 638
          032E 639
          032E 640
          032E 641
          032E 642
          032E 643
          032E 644
          032E 645
          032E 646
          032E 647
          032E 648
          032E 649
          032E 650
          032E 651
          032E 652
          032E 653
          032E 654
          032E 655
          032E 656
          032E 657
          032E 658
          032E 659
          032E 660
          032E 661
          032E 662
          032E 663
          032E 664
          032E 665
          032E 666
          032E 667
          032E 668
          032E 669
          032E 670
          032E 671
          032E 672
          032E 673
          032E 674
          032E 675
          032E 676
          032E 677
          032E 678
          032E 679
          032E 680
          032E 681
          032E 682
          032E 683
          032E 684
          032E 685
          032E 686
          032E 687
          032E 688
          032E 689
          032E 690
          032E 691
          032E 692
          032E 693
          032E 694
          032E 695
          032E 696
          032E 697
          032E 698
          032E 699
          032E 700
          032E 701
          032E 702
          032E 703
          032E 704
          032E 705
          032E 706
          032E 707
          032E 708
          032E 709
          032E 710
          032E 711
          032E 712
          032E 713
          032E 714
          032E 715
          032E 716
          032E 717
          032E 718
          032E 719
          032E 720
          032E 721
          032E 722
          032E 723
          032E 724
          032E 725
          032E 726
          032E 727
          032E 728
          032E 729
          032E 730
          032E 731
          032E 732
          032E 733
          032E 734
          032E 735
          032E 736
          032E 737
          032E 738
          032E 739
          032E 740
          032E 741
          032E 742
          032E 743
          032E 744
          032E 745
          032E 746
          032E 747
          032E 748
          032E 749
          032E 750
          032E 751
          032E 752
          032E 753
          032E 754
          032E 755
          032E 756
          032E 757
          032E 758
          032E 759
          032E 760
          032E 761
          032E 762
          032E 763
          032E 764
          032E 765
          032E 766
          032E 767
          032E 768
          032E 769
          032E 770
          032E 771
          032E 772
          032E 773
          032E 774
          032E 775
          032E 776
          032E 777
          032E 778
          032E 779
          032E 780
          032E 781
          032E 782
          032E 783
          032E 784
          032E 785
          032E 786
          032E 787
          032E 788
          032E 789
          032E 790
          032E 791
          032E 792
          032E 793
          032E 794
          032E 795
          032E 796
          032E 797
          032E 798
          032E 799
          032E 800
          032E 801
          032E 802
          032E 803
          032E 804
          032E 805
          032E 806
          032E 807
          032E 808
          032E 809
          032E 810
          032E 811
          032E 812
          032E 813
          032E 814
          032E 815
          032E 816
          032E 817
          032E 818
          032E 819
          032E 820
          032E 821
          032E 822
          032E 823
          032E 824
          032E 825
          032E 826
          032E 827
          032E 828
          032E 829
          032E 830
          032E 831
          032E 832
          032E 833
          032E 834
          032E 835
          032E 836
          032E 837
          032E 838
          032E 839
          032E 840
          032E 841
          032E 842
          032E 843
          032E 844
          032E 845
          032E 846
          032E 847
          032E 848
          032E 849
          032E 850
          032E 851
          032E 852
          032E 853
          032E 854
          032E 855
          032E 856
          032E 857
          032E 858
          032E 859
          032E 860
          032E 861
          032E 862
          032E 863
          032E 864
          032E 865
          032E 866
          032E 867
          032E 868
          032E 869
          032E 870
          032E 871
          032E 872
          032E 873
          032E 874
          032E 875
          032E 876
          032E 877
          032E 878
          032E 879
          032E 880
          032E 881
          032E 882
          032E 883
          032E 884
          032E 885
          032E 886
          032E 887
          032E 888
          032E 889
          032E 890
          032E 891
          032E 892
          032E 893
          032E 894
          032E 895
          032E 896
          032E 897
          032E 898
          032E 899
          032E 900
          032E 901
          032E 902
          032E 903
          032E 904
          032E 905
          032E 906
          032E 907
          032E 908
          032E 909
          032E 910
          032E 911
          032E 912
          032E 913
          032E 914
          032E 915
          032E 916
          032E 917
          032E 918
          032E 919
          032E 920
          032E 921
          032E 922
          032E 923
          032E 924
          032E 925
          032E 926
          032E 927
          032E 928
          032E 929
          032E 930
          032E 931
          032E 932
          032E 933
          032E 934
          032E 935
          032E 936
          032E 937
          032E 938
          032E 939
          032E 940
          032E 941
          032E 942
          032E 943
          032E 944
          032E 945
          032E 946
          032E 947
          032E 948
          032E 949
          032E 950
          032E 951
          032E 952
          032E 953
          032E 954
          032E 955
          032E 956
          032E 957
          032E 958
          032E 959
          032E 960
          032E 961
          032E 962
          032E 963
          032E 964
          032E 965
          032E 966
          032E 967
          032E 968
          032E 969
          032E 970
          032E 971
          032E 972
          032E 973
          032E 974
          032E 975
          032E 976
          032E 977
          032E 978
          032E 979
          032E 980
          032E 981
          032E 982
          032E 983
          032E 984
          032E 985
          032E 986
          032E 987
          032E 988
          032E 989
          032E 990
          032E 991
          032E 992
          032E 993
          032E 994
          032E 995
          032E 996
          032E 997
          032E 998
          032E 999
          032E 1000
          032E 1001
          032E 1002
          032E 1003
          032E 1004
          032E 1005
          032E 1006
          032E 1007
          032E 1008
          032E 1009
          032E 1010
          032E 1011
          032E 1012
          032E 1013
          032E 1014
          032E 1015
          032E 1016
          032E 1017
          032E 1018
          032E 1019
          032E 1020
          032E 1021
          032E 1022
          032E 1023
          032E 1024
          032E 1025
          032E 1026
          032E 1027
          032E 1028
          032E 1029
          032E 1030
          032E 1031
          032E 1032
          032E 1033
          032E 1034
          032E 1035
          032E 1036
          032E 1037
          032E 1038
          032E 1039
          032E 1040
          032E 1041
          032E 1042
          032E 1043
          032E 1044
          032E 1045
          032E 1046
          032E 1047
          032E 1048
          032E 1049
          032E 1050
          032E 1051
          032E 1052
          032E 1053
          032E 1054
          032E 1055
          032E 1056
          032E 1057
          032E 1058
          032E 1059
          032E 1060
          032E 1061
          032E 1062
          032E 1063
          032E 1064
          032E 1065
          032E 1066
          032E 1067
          032E 1068
          032E 1069
          032E 1070
          032E 1071
          032E 1072
          032E 1073
          032E 1074
          032E 1075
          032E 1076
          032E 1077
          032E 1078
          032E 1079
          032E 1080
          032E 1081
          032E 1082
          032E 1083
          032E 1084
          032E 1085
          032E 1086
          032E 108
```

```

00000001 0333      IF      IDN,<#0>,<#0>
00000001 0333      $ST1 = 1
00000001 0333      .ENDC
00000001 0333      .ENDC
00000001 0333      IF      NE      $ST1
7E    7C 0333      CLRQ   -(SP)
0335      IFF
0335      PUSHL  #0
0335      PUSHL  #0
0335      .ENDC
0335
00000000 0335      SPUSHWTW #0,#0
00000000 0335      $ST1 = 0
00000001 0335      .IF      IDN,<#0>,<#0>
00000001 0335      .IF      IDN,<#0>,<#0>
00000001 0335      $ST1 = 1
00000001 0335      .ENDC
00000001 0335      .ENDC
00000001 0335      IF      NE      $ST1
7E    7C 0335      CLRQ   -(SP)
0337      IFF
0337      PUSHL  #0
0337      PUSHL  #0
0337      .ENDC
0337
0000041A 8F  DD 0337      PUSHL  #MAXMSG
0000041A 8F  DD 033D      SPUSHADR AST$T_BUF(R5)
0000041A 8F  DD 033D      .IF      IDN,0,AST$T_BUF(R5)
0000041A 8F  DD 033D      PUSHL  #0
0000041A 8F  DD 033D      .IFF
0000041A 8F  DF 033D      PUSHAL  AST$T_BUF(R5)
0000041A 8F  DF 0340      .ENDC
0000041A 8F  DF 0340
00000000 0340      SQIOPUSH #0,0
00000000 0340      $ST1 = 0
00000001 0340      .IF      IDN,<#0>,<#0>
00000001 0340      .IF      IDN,<0>,<0>
00000001 0340      $ST1 = 1
00000001 0340      .ENDC
00000001 0340      .ENDC
00000001 0340      IF      NE $ST1
7E    7C 0340      CLRQ   -(SP)
0342      IFF
0342      PUSHL  #0
0342      SPUSHADR 0
0342      .ENDC
0342
04 A5  7F 0342      SPUSHADR AST$Q_IOSB(R5),CONTEXT=Q
04 A5  7F 0342      .IF      IDN,0,AST$Q_IOSB(R5)
04 A5  7F 0342      PUSHL  #0
04 A5  7F 0342      .IFF
04 A5  7F 0345      PUSHAQ  AST$Q_IOSB(R5)
04 A5  7F 0345      .ENDC
7E    7E  31  3C 0345      MOVZWL #IOS_READVBLK,-(SP)
7E    0144 CF  3C 0348      MOVZWL LINKCHAN,-(SP)
7E    00  DD  0348      PUSHL  #0

```

```

00000000'GF 0C FB 034F          CALLS #12,G^SYSSQIOW
                0356
                0356
17 50 E8 0356 494 100$: BLBS R0, 110$ ; Branch if ok
      50 DD 0359 495           PUSHL R0 ; Save error status
                035B 496           $PUTMSG_S - ; Output an error message
                035B 497           MSGVEC = DECNETERR ; saying some sort of link error
                035B
                035B
00 DD 035B 035D PUSHL #0
                035D $PUSHADR 0,CONTEXT=Q
                035D .IF IDN,0,0
00 DD 035D 035F PUSHL #0
                035F .IFF
                035F PUSHAQ 0
                035F .ENDC
                035F
                035F $PUSHADR 0
                035F .IF IDN,0,0
00 DD 035F 0361 PUSHL #0
                0361 .IFF
                0361 PUSHAL 0
                0361 .ENDC
                0361
                0361 $PUSHADR DECNETERR
                0361 .IF IDN,0,DECNETERR
                0361 PUSHL #0
                0361 .IFF
                0361 PUSHAL DECNETERR
                0361 .ENDC
0110'CF DF 0361 0365
                0365
                0365 CALLS #4,G^SYSSPUTMSG
                036C
                036C
                50 8ED0 036C 498 498 POPL R0 ; Restore the error status
                04 036F 499           RET ; and die
                0370 500
                50 04 A5 3C 0370 501 110$: MOVZWL AST$Q_IOSB(R5), R0 ; Get the I/O completion code
                E2 50 E9 0374 502           BLBC R0, 100$ ; Go die unless success completion
                0377 503
                0377 504 : ***** start temp old RSTS/E *****
                0377 505
                50 06 A5 01 A3 0377 506 SUBW3 #1, AST$Q_IOSB+2(R5), R0
                27 A5 50 B1 037C 507 CMPW R0, AST$T_BUF+1(R5)
                10 12 0380 508 BNEQ FOO_RSTS_T
                26 A5 00000101 8F DO 0382 509 MOVL #1@8!1, AST$T_BUF(R5)
                2A A5 00010002 8F DO 038A 510 MOVL #1@0@16!2, AST$T_BUF+4(R5)
                00000392 0392 511 FOO_RSTS_1 = .
                0392 512
                0392 513 : ***** end temp old RSTS/E *****
                0392 514
                01 26 A5 91 0392 515 CMPB AST$T_BUF(R5), #1 ; Is it a CONFIG message?
                06 12 0396 516 BNEQ 130$ ; Nope
                01 27 A5 91 0398 517 CMPB AST$T_BUF+1(R5), #1 ; A CONFIG message for V1 or higher?
                12 18 039C 518 BGEQ 140$ ; Yep
                039E 519 130$: $PUTMSG_S - ; Output an error message
                039E 520           MSGVEC = NOTVMS ; saying protocol not supported
                039E
                00 DD 039E 03A0           GLOBL SYSSPUTMSG
                039E           PUSHL #0
                039E           $PUSHADR 0,CONTEXT=Q

```

```

00 DD 03A0      .IF IDN,0,0
                PUSHL #0
                IFF
                PUSHAQ 0
                .ENDC

00 DD 03A2      $PUSHADR 0
                .IF IDN,0,0
                PUSHL #0
                IFF
                PUSHAL 0
                .ENDC

0138'CF DF 03A4      $PUSHADR NOTVMS
                .IF IDN,0,NOTVMS
                PUSHL #0
                IFF
                PUSHAL NOTVMS
                .ENDC

00000000'GF 04 FB 03A8      CALLS #4,G^SYSSPUTMSG
                03A8
                03A8
                03AF      521      RET
                03B0      522
                03B0      523
                03B0      524      : Use the support bit mask that was returned to determine which
                03B0      525      : of the protocol modules to call.
                03B0      526
                03B0      527

0319'CF 28 A5 90 03B0      140$: MOVB AST$T_BUF+2(R5),PROTO_ECO ; Save eco level
031A'CF 2A A5 B0 03B6      MOVW AST$T_BUF+4(R5),HOST_OPSYS ; Save host operating system
                03BC      528
                03BC      529
                03C1      530
                03C1      531      150$: MOVAB PROTOTBL-4, R2          : Get (biased) pointer to protocols
                03C1      532
                03C1      533      ADDL #4, R2
                03C4      534      CMPL R2,#ENDPROTO      : Index over address to bit number
                03CB      535      BGEOU 130$           : Are we out of protocols?
                03CD      536      BITW AST$T_BUF+6(R5),(R2)+   : Yep, protocol not supported
                03D1      537      BEQL 150$           : Not yet, does the bit match?
                03D3      538      : No match, loop for next protocol
                03D3      539
                03D3      540      : R2 is now address of protocol module routine
                03D3      541
                03D7      542
                03DB      543      PUSHAL OLDCTRL      : Arg #2 is returned out-of-band bits
                03E2      544      PUSHAL OLDCTRL      : Arg #1 is out-of-band's to disable
                03E2      545      CALLS #2,G^LIB$DISABLE_CTRL ; Get the currently enabled ones
                03E6      546
                03EA      547      PUSHAL OLDCTRL      : Arg #2 is returned out-of-band bits
                03F1      548      PUSHAL OLDCTRL      : Arg #1 is out-of-band's to disable
                03F1      549      CALLS #2,G^LIB$DISABLE_CTRL ; Go disable out-of-bound ASTs
                03F1      550      $SETRWM_S -
                C3F1      550      -WATFLG = #1      : Set resource wait mode
                03F1      550      .GLOBL SYSSSETRWM    : off so we never will hang
                03F1      551      PUSHL #1
                03F3      551      CALLS #1,G^SYSSSETRWM
                03FA      551      CMPW R0, #SSS_WASCLR    : Was it already off?

0000'BF 50 B1 03FA      551

```



```

0208'CF 7F 0455 584      PUSHAQ  PSTHRU MSG          ; Arg #1 is the string to append to
00000000'GF 02 FB 0459 585      CALLS   #2, G^STR$APPEND    Gc save message by appending
                                50 D4 0460 586      CLRL    R0               Say we don't want any output now
                                04 0462 587      RET                ; then return
                                0463 588
                                0463 589 CTERM_RT:           ; CTERM protocol initialization
0000 0463 590      .WORD  ^M<>          ; No register(s) to save
                                0465 591
02B2'CF 96 0465 592      INCB    CTERM_FLAG        ; Indicate CTERM protocol
                                0469 593
                                04E9'CF 7E 0469 594      movaq   infomsg1,RO      ; assume not vax to vax
031A'CF 07 B1 046E 595      CMPW   #7 HOST_OPSYS    Talking to VMS?
                                0A 12 0473 596      BNEQ   10$              ; nope
                                10 A8 0475 597      BISW   #FLGSM VAXHOST,-
02B2'CF 0477 598      CTERM_FLAG        ; Set flag
0522'CF 7E 047A 599      movaq   infomsg2,RO      ; vax to vax
                                047F 600 10$:           ; Branch if not requested
                                00 E1 047F 601      BBC    #RTLOG$V BANNER,-
11 02B6'CF 0481 602      RTLOG_FLAGS,20$       ; Branch if not requested
                                50 DD 0485 603      pushl   R0
                                01 DD 0487 604      pushl   #1
01FE1130'8F DD 0489 605      pushl   #<shr$text:sts$k_info>!<rem$facility@16> :
00000000'GF 03 FB 048F 606      calls   #3,g^lib$signal  ;
                                0496 607
                                0496 608 20$:           ; Go join the VMSRT protocol
                                049B 609      CALLS   #0, VMSRT        then exit

```

030D'CF 6E DD 049C 611 .SBTTL RECORD_QUIT - snapshot QUIT info
049C 612
049C 613 RECORD_QUIT:::
049C 614
030D'CF 6E DD 049C 615 MOVL (SP),QUIT_PC ; save caller PC
04A1 616
04A1 617 \$GETJPIW S -
04A1 618 EFN = #5,-
04A1 619 PIDADR = LOCAL_PID,-
04A1 620 ITMLST = GETJPI_ITMLST,-
04A1 621 IOSB = JPI_IOSB
04A1 .GLOBL SYSS\$GETJPIW
04A1 \$QIOPUSH #0,0
00000000 04A1 \$\$T1 = 0
04A1 .IF IDN,<#0>,<#0>
04A1 .IF IDN,<0>,<0>
00000001 04A1 \$\$T1 = 1
04A1 .ENDC
04A1 .ENDC
00000001 04A1 .IF NE \$\$T1
7E 7C 04A1 CLRQ -(SP)
04A3 .IFF
04A3 PUSHL #0
04A3 \$PUSHADR 0
04A3 .ENDC
04A3 .PUSHADR JPI_IOSB,CONTEXT=Q
04A3 .IF IDN,0,JPI_IOSB
04A3 PUSHL #0
04A3 .IFF
0004'CF 7F 04A3 PUSHAQ JPI_IOSB
04A7 .ENDC
04A7 .PUSHADR GETJPI_ITMLST
04A7 .IF IDN,0,GETJPI_ITMLST
04A7 PUSHL #0
04A7 .IFF
000C'CF DF 04A7 PUSHAL GETJPI_ITMLST
04AB .ENDC
04AB .PUSHADR 0,CONTEXT=Q
00 DD 04AB .IF IDN,0,0
04AD PUSHL #0
04AD .IFF
04AD PUSHAQ 0
04AD .ENDC
04AD .PUSHADR LOCAL_PID
04AD .IF IDN,0,[OCAL_PID
04AD PUSHL #0
04AD .IFF
0000'CF DF 04AD PUSHAL LOCAL_PID
04B1 .ENDC
04B1 .PUSHL #5
00000000'GF 05 DD 04B1 CALLS #7,G^SYSS\$GETJPIW
04B3
04BA

RTPAD
V04-000

- REMOTE TERMINAL PROGRAM
RECORD_QUIT - snapshot QUIT info

D 1

16-SEP-1984 02:15:27 VAX/VMS Macro V04-00
5-SEP-1984 03:15:47 [RTPAD.SRC]RTPAD.MAR;1

Page 34
(2)

50 08 50 E9 04BA 622 BLBC R0,100\$
0004'CF 3C 04BD 623 MOVZWL JPI IOSB,R0
00 50 E9 04C2 624 BLBC R0,T00\$
04C5 625 100\$: RSB
05 04C5 626
04C6 627

RTP
V04

4F 4E 24 53 59 53 000004CE'010E0000' 04C6 629 .SBTTL READ ONLY DATA
 45 44 04C6 630 SYSSNODE: .ASCID /SYSSNODE/
 4F 43 24 53 59 53 000004DE'010E0000' 04D4 631 TTYDESC:: .ASCID /SYS\$COMMAND/
 44 4E 41 4D 4D 04D6 632
 04E4 633 infomsg1: .ascid /remote terminal is using CTERM protocol (non-vax)/ ; *** TE
 04E9 634
 65 74 6F 6D 65 72 000004F1'010E0000' 73 69 20 6C 61 6E 69 6D 72 65 74 20 04F7
 04F7 635 infomsg2: .ascid /remote terminal is using CTERM protocol (vax-to-vax)/ ; ***
 0503 636 DVILIST:
 4D 52 45 54 43 20 67 6E 69 73 75 20 0503 637 0004 0004 0522
 0530 638 .WORD 4,DVI\$ DEVCLASS
 76 22 20 6C 6F 63 6F 74 6F 72 70 20 053C 639 .ADDRESS DÉVCLASS_TEMP,0 ; Device class
 0548 640
 0554 641 .WORD 4,DVI\$ DEVTYPE
 0554 642 .ADDRESS DÉVTYPÉ_TEMP,0 ; Device type
 055E 643
 055E 644 .WORD 4,DVI\$ DEVBUFSIZ
 055E 645 .ADDRESS DÉVBUFSIZ_TEMP,0 ; Device buffer size
 055E 646
 055E 647 .WORD 4,DVI\$ DEVDEPEND
 055E 648 .ADDRESS DÉVDEPEND,0 ; Device dependant data (1)
 055E 649
 055E 650 .WORD 4,DVI\$ DEVDEPEND2
 055E 651 .ADDRESS DÉVDEPEND2,0 ; Device dependant data (2)
 055E 652
 055E 653 .WORD 16,DVI\$ DEVNAM
 055E 654 .ADDRESS DEVNAM,DEVNAMLEN ; Device name and length
 05A6 655
 05A6 656 .WORD 4,DVI\$ UNIT
 05A6 657 .ADDRESS TERMUNIT,0 ; Device unit number
 05B2 658
 05B2 659 .LONG 0 :END OF LIST
 05B2 660

```

      05B6   662     .SBTTL READ WRITE DATA
      00000000 663     .PSECT _RTPAD, LONG
      0000 664
      00000000 0000 665 LOCAL PID: .LONG 0 ; no pid
      0000000C 0004 666 JPI_IOSB: .BLKL 2 ; iosb
      000C 667
      000C 668 GETJPI_ITMLST:
      0000'0004 000C 669 .WORD 4, JPI$ ASTLM
      00000050'00000040' 0010 670 .LONG ASTLMLEN
      0000'0004 0018 671 .WORD 4, JPI$ ASTCNT
      00000050'00000044' 001C 672 .LONG ASTCNTLEN
      0000'0004 0024 673 .WORD 4, JPI$ BIOLM
      00000050'00000048' 0028 674 .LONG BIOLMLEN
      0000'0004 0030 675 .WORD 4, JPI$ BIOCNT
      00000050'0000004C' 0034 676 .LONG BIOCNTLEN
      003C 677
      00000000 003C 678 .LONG 0 ; end of list
      0040 679
      00000000 0040 680 ASTLM: .LONG 0
      00000000 0044 681 ASTCNT: .LONG 0
      00000000 0048 682 BIOLM: .LONG 0
      00000000 004C 683 BIOCNT: .LONG 0
      00000000 0050 684 LEN: .LONG 0
      0054 685
      0054 686 :
      0054 687 ; Returned data area for $GETDVI
      0054 688 :
      0054 689 :
      00000000 0054 690 DEVCLASS TEMP: .LONG 0 ; Temp locations since getdvi wants longs
      00000000 0058 691 DEVTYPE TEMP: .LONG 0
      00000000 005C 692 DEVBUFSIZ TEMP: .LONG 0
      0060 693
      0060 694 CHAR BLOCK:: ; VMS characteristics
      00000061 0060 695 DEVCLASS: .BLKB 1
      00000062 0061 696 DEVTYPE: .BLKB 1
      00000064 0062 697 DEVBUFSIZ: .BLKW 1
      00000068 0064 698 DEVDEPEND: .BLKL 1
      0000006C 0068 699 DEVDEPEND2: .BLKL 1
      006C 700
      00000000 006C 701 DEVNAMLEN:: .LONG 0 ;
      0070 702
      00000080 0070 703 DEVNAM:: .BLKB 16 ;
      0080 704
      00000000 0080 705 TERMUNIT:: .LONG 0 ;
      0084 706
      0084 707 TERMCHAR:: ; Term characteristics for RSX, etc.
      0000008C'00000084 0084 708 .LONG DIB$K_LENGTH+16,1$ ; Term characteristics for RSX, etc.
      00000110 008C 709 1$: .BLKB DIB$K_LENGTH+16
      0110 710
      0110 711 :
      0110 712 ; Message vectors for $PUTMSG
      0110 713 :
      0110 714
      00000000 00000000'00000002 0110 715 DECNETERR:: .LONG 2, REM$_NETERR,0
      011C 716
      00000000 00000000'00000002 011C 717 NOTTERM: .LONG 2, REM$_NOTERM,0
      0128 718

```

- REMOTE TERMINAL PROGRAM
READ WRITE DATA

G 1

16-SEP-1984 02:15:27 VAX/VMS Macro V04-00
5-SEP-1984 03:15:47 [RTPAD.SRC]RTPAD.MAR;1Page 37
(3)

000001B0'00000001 00000000'00000003 0128 719 EXITMSG: .LONG 3,REM\$_END,1,NODENAME
0138 720
00000000 00000000'00000002 0138 721 NOTVMS: .LONG 2,REM\$_NOPROT,0
0144 722
0144 723 :
0144 724 : Channels assigned in INIT and used by other modules
0144 725 :
00000000 0144 726
0148 727 LINKCHAN::: .LONG 0 ; DECnet link channel
00000000 0148 728 MAILCHAN::: .LONG 0 ; DECnet link mailbox channel
014C 730
014C 731 RDWRTCHAN:::
00000000 014C 732 READCHAN::: .LONG 0 ; Terminal reads channel
0150 733
00000000 0150 734 WRITECHAN::: .LONG 0 ; Terminal writes channel
0154 735
00000000 0154 736 CNTRLCHAN::: .LONG 0 ; Terminal ^C and ^Y enables
0158 737
00000000 0158 738 TERMMBXCHAN::: .LONG 0 ; Terminal Unsolicited data mailbox channel
015C 739

```

015C 741 ;  

015C 742 : Cli interface storage  

015C 743:  

44 4C 4F 00000164'010E0000' 015C 744 OLD_DESC: .ASCID /OLD/ ; /OLD qualifier  

0167 745  

47 4F 4C 0000016F'010E0000' 0167 746 LOG_DESC: .ASCID /LOG/ ; /LOG qualifier  

0172 747  

45 54 44 0000017A'010E0000' 0172 748 DTE_DESC: .ASCID /DTE/ ; /DTE qualifier  

017D 749  

45 44 4F 4E 00000185'010E0000' 017D 750 NODEDESC: .ASCID /NODE/ ; name of parameter  

0189 751  

0189 752:  

0189 753 : Network names and descriptors  

0189 754:  

0189 755  

0189 756 OBJ_DESC:  

00000007' 0189 757 .LONG 20$-10$ ; Object type descriptor  

00000191' 018D 758 .LONG 10$  

0191 759  

22 3A 3A 0191 760 10$: .ASCII /::'/' ; FINAL FORMAT: node_name::'nn=''  

00000003 0194 761 OBJ_C_PREFIX = .-10$ ; Offset to object number  

22 3D 32 34 0194 762 .ASCII /42='/' ; 42 is TSA, 23 is old remote terminal  

0198 763 20$:  

0198 764  

0198 765 NODE_NAME_DESC:  

02000000 0198 766 .LONG DSC$K_CLASS_DA<DSC$B_CLASS*8> ; Returned CLI node name parameter  

00000000 019C 767 .LONG 0  

01A0 768 LOG_FILE_DESC:  

02000000 01A0 769 .LONG DSC$K_CLASS_DA<DSC$B_CLASS*8> ; Returned /LOG= file spec  

00000000 01A4 770 .LONG 0  

01A8 771  

01A8 772 CONNDESC:  

02000000 01A8 773 .LONG DSC$K_CLASS_DA<DSC$B_CLASS*8> ; Network connection string  

00000000 01AC 774 .LONG 0  

01B0 775  

00000040' 01B0 776 NODENAME:: .LONG 20$-10$ ; Translation of SYS$NODE held here  

000001B8' 01B4 777 .LONG 10$  

01B8 778  

000001F8 01B8 779 10$: .BLKB 64  

01F8 780 20$:  

01F8 781  

01F8 782 FINALPATH:: .LONG DSC$K_CLASS_DA<DSC$B_CLASS*8> ; Final path descriptor  

02000000 01F8 783 .LONG 0  

00000000 01FC 784 .LONG 0  

0200 785  

02000000 0200 786 FINALACS:: .LONG DSC$K_CLASS_DA<DSC$B_CLASS*8> ; Final ACS descriptor  

00000000 0204 787 .LONG 0  

0208 788  

0208 789  

0208 790 PSTHRU_MSG: .LONG DSC$K_CLASS_DA<DSC$B_CLASS*8> ; Saved PSTHRU message descriptor  

0208 791 .LONG 0  

00000000 020C 792 .LONG 0  

0210 793  

0A 0D 00000218'010E0000' 0210 794 PSTHRU_CRLF: .ASCID <13><10> ; <CR><LF> for PSTHRU messages  

021A 795  

021A 796:  

021A 797 ; RMS storage

```

```
021A 798 :  
021A 799 : .ALIGN LONG  
021C 800  
021C 801 SYSINFAB:: $FAB FAC=GET,FNM=<SYSS$INPUT> ; To open SYSS$INPUT  
021C $FABDEF  
021C $DEFINI FAB,  
021C .SAVE LOCAL_BLOCK  
021C .NOCROSS  
021C .IIF DIF <> <GLOBAL>,ENABLE SUPPRESSION  
021C .PSECT $ABS$,ABS  
0000 $GBLINI  
0000 .IF IDN <LOCAL> <GLOBAL>  
0000 .MACRO $DEF SYM,ALLOC,SIZ  
0000 .IIF NB,SYM, SYM:  
0000 .IIF NB,ALLOC, ALLOC SIZ  
0000 .ENDM $DEF  
0000 .MACRO SEQU SYM,VAL  
0000 SYM==VAL  
0000 .ENDM SEQU  
0000 .MACRO $VIELD1 MOD,SEP,SYM,SIZ,MSK  
0000 SIZ...=1  
0000 .IIF NB,SIZ, SIZ...=SIZ  
0000 .IF NB,SYM  
0000 MOD'SEP'V 'SYM==BIT..  
0000 .IIF NB,SIZ, MOD'SEP'S 'SYM==SIZ  
0000 .IIF NB,MSK, MOD'SEP'M_ 'SYM==<<<1@SIZ...>-1>@BIT...>  
0000 .ENDC  
0000 BIT...=BIT...+SIZ...  
0000 .ENDM $VIELD1  
0000 .IFF  
0000 .IIF DIF <LOCAL> <LOCAL>, ERROR ;ARG MUST BE "GLOBAL","LOCAL",OR NULL  
0000 .MACRO $DEF SYM,ALLOC,SIZ  
0000 .IIF NB,SYM, SYM:  
0000 .IIF NB,ALLOC, ALLOC SIZ  
0000 .ENDM $DEF  
0000 .MACRO SEQU SYM,VAL  
0000 SYM=VAL  
0000 .ENDM SEQU  
0000 .MACRO $VIELD1 MOD,SEP,SYM,SIZ,MSK  
0000 SIZ...=1  
0000 .IIF NB,SIZ, SIZ...=SIZ  
0000 .IF NB,SYM  
0000 MOD'SEP'V 'SYM==BIT..  
0000 .IIF NB,SIZ, MOD'SEP'S 'SYM=SIZ  
0000 .IIF NB,MSK, MOD'SEP'M_ 'SYM=<<<1@SIZ...>-1>@BIT...>  
0000 .ENDC  
0000 BIT...=BIT...+SIZ...  
0000 .ENDM $VIELD1  
0000 .ENDC  
00000000 0000 .=0  
00000003 0000 SEQU FAB$C_BID 3  
00000003 0000 FAB$C_BID=3  
00003FC0 0000 SEQU FAB$M_PPF_RAT 16320  
00003FC0 0000 FAB$M_PPF_RAT=16320
```

00004000	0000	SEQU	FAB\$M_PPF_IND	16384
			FAB\$M_PPF_IND=	16384
00000002	0000	SEQU	FAB\$M_MXV	2
			FAB\$M_MXV=	2
00000004	0000	SEQU	FAB\$M_SUP	4
			FAB\$M_SUP=	4
00000008	0000	SEQU	FAB\$M_TMP	8
			FAB\$M_TMP=	8
00000010	0000	SEQU	FAB\$M_TMD	16
			FAB\$M_TMD=	16
00000020	0000	SEQU	FAB\$M_DFW	32
			FAB\$M_DFW=	32
00000040	0000	SEQU	FAB\$M_SQO	64
			FAB\$M_SQO=	64
00000080	0000	SEQU	FAB\$M_RWO	128
			FAB\$M_RWO=	128
00000100	0000	SEQU	FAB\$M_POS	256
			FAB\$M_POS=	256
00000200	0000	SEQU	FAB\$M_WCK	512
			FAB\$M_WCK=	512
00000400	0000	SEQU	FAB\$M_NEF	1024
			FAB\$M_NEF=	1024
00000800	0000	SEQU	FAB\$M_RWC	2048
			FAB\$M_RWC=	2048
00001000	0000	SEQU	FAB\$M_DMO	4096
			FAB\$M_DMO=	4096
00002000	0000	SEQU	FAB\$M_SPL	8192
			FAB\$M_SPL=	8192
00004000	0000	SEQU	FAB\$M_SCF	16384
			FAB\$M_SCF=	16384
00008000	0000	SEQU	FAB\$M_DLT	32768
			FAB\$M_DLT=	32768
00010000	0000	SEQU	FAB\$M_NFS	65536
			FAB\$M_NFS=	65536
00020000	0000	SEQU	FAB\$M_UFO	131072
			FAB\$M_UFO=	131072
00040000	0000	SEQU	FAB\$M_PPF	262144
			FAB\$M_PPF=	262144

00080000	0000	SEQU	FAB\$M_INP 524288 FAB\$M_INP=524288
00100000	0000	SEQU	FAB\$M_CTG 1048576 FAB\$M_CTG=1048576
00200000	0000	SEQU	FAB\$M_CBT 2097152 FAB\$M_CBT=2097152
00800000	0000	SEQU	FAB\$M_RCK 8388608 FAB\$M_RCK=8388608
01000000	0000	SEQU	FAB\$M_NAM 16777216 FAB\$M_NAM=16777216
02000000	0000	SEQU	FAB\$M_CIF 33554432 FAB\$M_CIF=33554432
08000000	0000	SEQU	FAB\$M_ESC 134217728 FAB\$M_ESC=134217728
10000000	0000	SEQU	FAB\$M_TEF 268435456 FAB\$M_TEF=268435456
20000000	0000	SEQU	FAB\$M_OFP 536870912 FAB\$M_OFP=536870912
40000000	0000	SEQU	FAB\$M_KFO 1073741824 FAB\$M_KFO=1073741824
00000001	0000	SEQU	FAB\$M_PUT 1 FAB\$M_PUT=1
00000002	0000	SEQU	FAB\$M_GET 2 FAB\$M_GET=2
00000004	0000	SEQU	FAB\$M_DEL 4 FAB\$M_DEL=4
00000008	0000	SEQU	FAB\$M_UPD 8 FAB\$M_UPD=8
00000010	0000	SEQU	FAB\$M_TRN 16 FAB\$M_TRN=16
00000020	0000	SEQU	FAB\$M_BIO 32 FAB\$M_BIO=32
00000040	0000	SEQU	FAB\$M_BRO 64 FAB\$M_BRO=64
00000080	0000	SEQU	FAB\$M_EXE 128 FAB\$M_EXE=128
00000001	0000	SEQU	FAB\$M_SHRPUT 1 FAB\$M_SHRPUT=1

00000002	0000	SEQU	FAB\$M_SHRGET FAB\$M_SHRGET=2	2
00000004	0000	SEQU	FAB\$M_SHRDEL FAB\$M_SHRDEL=4	4
00000008	0000	SEQU	FAB\$M_SHRUPD FAB\$M_SHRUPD=8	8
00000010	0000	SEQU	FAB\$M_MSE FAB\$M_MSE=16	16
00000020	0000	SEQU	FAB\$M_NIL FAB\$M_NIL=32	32
00000040	0000	SEQU	FAB\$MUPI FAB\$MUPI=64	64
00000000	0000	SEQU	FAB\$C_SEQ FAB\$C_SEQ=0	0
00000010	0000	SEQU	FAB\$C_REL FAB\$C_REL=16	16
00000020	0000	SEQU	FAB\$C_IDX FAB\$C_IDX=32	32
00000030	0000	SEQU	FAB\$C_HSH FAB\$C_HSH=48	48
00000001	0000	SEQU	FAB\$M_FTN FAB\$M_FTN=1	1
00000002	0000	SEQU	FAB\$M_CR FAB\$M_CR=2	2
00000004	0000	SEQU	FAB\$M_PRN FAB\$M_PRN=4	4
00000008	0000	SEQU	FAB\$M_BLK FAB\$M_BLK=8	8
00000002	0000	SEQU	FAB\$C_RFMDFLT FAB\$C_RFMDFLT=2	2
00000000	0000	SEQU	FAB\$C_UDF FAB\$C_UDF=0	0
00000001	0000	SEQU	FAB\$C_FIX FAB\$C_FIX=1	1
00000002	0000	SEQU	FAB\$C_VAR FAB\$C_VAR=2	2
00000003	0000	SEQU	FAB\$C_VFC FAB\$C_VFC=3	3

00000004	0000	SEQU	FAB\$C_STM FAB\$C_STM=4	4
00000005	0000	SEQU	FAB\$C_STMLF FAB\$C_STMLF=5	5
00000006	0000	SEQU	FAB\$C_STMCR FAB\$C_STMCR=6	6
00000006	0000	SEQU	FAB\$C_MAXRFM FAB\$C_MAXRFM=6	6
00000001	0000	SEQU	FAB\$M_RU FAB\$M_RU=1	1
00000002	0000	SEQU	FAB\$M_AI FAB\$M_AI=2	2
00000004	0000	SEQU	FAB\$M_BI FAB\$M_BI=4	4
00000050	0000	SEQU	FAB\$K_BLN FAB\$K_BLN=80	80
00000050	0000	SEQU	FAB\$C_BLN FAB\$C_BLN=80	80
00000050	0000	SEQU	FAB\$S_FABDEF FAB\$S_FABDEF=80	80
00000000	0000	SEQU	FAB\$B_BID FAB\$B_BID=0	0
00000001	0000	SEQU	FAB\$B_BLN FAB\$B_BLN=1	1
00000002	0000	SEQU	FAB\$RIFI_OVERLAY FAB\$RIFI_OVERLAY=2	2
00000002	0000	SEQU	FAB\$WIFI FAB\$WIFI=2	2
00000002	0000	SEQU	FAB\$RIFI_BITS FAB\$RIFI_BITS=2	2
00000008	0000	SEQU	FAB\$S_PPF_RAT FAB\$S_PPF_RAT=8	8
00000006	0000	SEQU	FAB\$V_PPF_RAT FAB\$V_PPF_RAT=6	6
0000000E	0000	SEQU	FAB\$V_PPF_IND FAB\$V_PPF_IND=14	14
00000004	0000	SEQU	FAB\$R_FOP_OVERLAY FAB\$R_FOP_OVERLAY=4	4

00000004	0000	SEQU	FAB\$L_FOP FAB\$L_FOP=4	4
00000004	0000	SEQU	FAB\$R_FOP_BITS FAB\$R_FOP_BITS=4	
00000001	0000	SEQU	FAB\$V_MXV FAB\$V_MXV=1	1
00000002	0000	SEQU	FAB\$V_SUP FAB\$V_SUP=2	2
00000003	0000	SEQU	FAB\$V_TMP FAB\$V_TMP=3	3
00000004	0000	SEQU	FAB\$V_TMD FAB\$V_TMD=4	4
00000005	0000	SEQU	FAB\$V_DFW FAB\$V_DFW=5	5
00000006	0000	SEQU	FAB\$V_SQO FAB\$V_SQO=6	6
00000007	0000	SEQU	FAB\$V_RWO FAB\$V_RWO=7	7
00000008	0000	SEQU	FAB\$V_POS FAB\$V_POS=8	8
00000009	0000	SEQU	FAB\$V_WCK FAB\$V_WCK=9	9
0000000A	0000	SEQU	FAB\$V_NEF FAB\$V_NEF=10	10
0000000B	0000	SEQU	FAB\$V_RWC FAB\$V_RWC=11	11
0000000C	0000	SEQU	FAB\$V_DMO FAB\$V_DMO=12	12
0000000D	0000	SEQU	FAB\$V_SPL FAB\$V_SPL=13	13
0000000E	0000	SEQU	FAB\$V_SCF FAB\$V_SCF=14	14
0000000F	0000	SEQU	FAB\$V_DLT FAB\$V_DLT=15	15
00000010	0000	SEQU	FAB\$V_NFS FAB\$V_NFS=16	16
00000011	0000	SEQU	FAB\$V_UFO FAB\$V_UFO=17	17

00000012	0000	SEQU	FAB\$V_PPF FAB\$V_PPF=18	18
00000013	0000	SEQU	FAB\$V_INP FAB\$V_INP=19	19
00000014	0000	SEQU	FAB\$V_CTG FAB\$V_CTG=20	20
00000015	0000	SEQU	FAB\$V_CBT FAB\$V_CBT=21	21
00000017	0000	SEQU	FAB\$V_RCK FAB\$V_RCK=23	23
00000018	0000	SEQU	FAB\$V_NAM FAB\$V_NAM=24	24
00000019	0000	SEQU	FAB\$V_CIF FAB\$V_CIF=25	25
0000001B	0000	SEQU	FAB\$V_ESC FAB\$V_ESC=27	27
0000001C	0000	SEQU	FAB\$V_TEF FAB\$V_TEF=28	28
0000001D	0000	SEQU	FAB\$V_OFP FAB\$V_OFP=29	29
0000001E	0000	SEQU	FAB\$V_KFO FAB\$V_KFO=30	30
00000008	0000	SEQU	FAB\$L_STS FAB\$L_STS=8	8
0000000C	0000	SEQU	FAB\$L_STV FAB\$L_STV=12	12
00000010	0000	SEQU	FAB\$L_ALQ FAB\$L_ALQ=16	16
00000014	0000	SEQU	FAB\$W_DEQ FAB\$W_DEQ=20	20
00000016	0000	SEQU	FAB\$R_FAC_OVERLAY FAB\$R_FAC_OVERLAY=22	22
00000016	0000	SEQU	FAB\$B_FAC FAB\$B_FAC=22	22
00000016	0000	SEQU	FAB\$R_FAC_BITS FAB\$R_FAC_BITS=22	22
00000000	0000	SEQU	FAB\$V_PUT FAB\$V_PUT=0	0

- REMOTE TERMINAL PROGRAM
READ WRITE DATA

C 2

16-SEP-1984 02:15:27 VAX/VMS Macro V04-00
5-SEP-1984 03:15:47 [RTPAD.SRC]RTPAD.MAR;1Page 46
(3)

	0000	SEQU	FAB\$V_GET FAB\$V_GET=1	1
00000001	0000	SEQU	FAB\$V_DEL FAB\$V_DEL=2	2
00000002	0000	SEQU	FAB\$V_UPD FAB\$V_UPD=3	3
00000003	0000	SEQU	FAB\$V_TRN FAB\$V_TRN=4	4
00000004	0000	SEQU	FAB\$V_BIO FAB\$V_BIO=5	5
00000005	0000	SEQU	FAB\$V_BRO FAB\$V_BRO=6	6
00000006	0000	SEQU	FAB\$V_EXE FAB\$V_EXE=7	7
00000007	0000	SEQU	FAB\$R_SHR_OVERLAY FAB\$R_SHR_OVERLAY=23	23
00000017	0000	SEQU	FAB\$B_SHR FAB\$B_SHR=23	23
00000017	0000	SEQU	FAB\$R_SHR_BITS FAB\$R_SHR_BITS=23	23
00000000	0000	SEQU	FAB\$V_SHRPUT FAB\$V_SHRPUT=0	0
00000001	0000	SEQU	FAB\$V_SHRGET FAB\$V_SHRGET=1	1
00000002	0000	SEQU	FAB\$V_SHRDEL FAB\$V_SHRDEL=2	2
00000003	0000	SEQU	FAB\$V_SHRUPD FAB\$V_SHRUPD=3	3
00000004	0000	SEQU	FAB\$V_MSE FAB\$V_MSE=4	4
00000005	0000	SEQU	FAB\$V_NIL FAB\$V_NIL=5	5
00000006	0000	SEQU	FAB\$VUPI FAB\$VUPI=6	6
00000018	0000	SEQU	FAB\$L_CTX FAB\$L_CTX=24	24
0000001C	0000	SEQU	FAB\$B_RTV FAB\$B_RTV=28	28

	0000	0000	SEQU	FAB\$R_ORG_OVERLAY FAB\$R_ORG_OVERLAY=29	29
0000001D	0000	0000	SEQU	FAB\$B_ORG FAB\$B_ORG=29	29
0000001D	0000	0000	SEQU	FAB\$R_ORG_BITS FAB\$R_ORG_BITS=29	29
00000004	0000	0000	SEQU	FAB\$S_ORG FAB\$S_ORG=4	4
00000004	0000	0000	SEQU	FAB\$V_ORG FAB\$V_ORG=4	4
0000001E	0000	0000	SEQU	FAB\$R_RAT_OVERLAY FAB\$R_RAT_OVERLAY=30	30
0000001E	0000	0000	SEQU	FAB\$B_RAT FAB\$B_RAT=30	30
0000001E	0000	0000	SEQU	FAB\$R_RAT_BITS FAB\$R_RAT_BITS=30	30
00000000	0000	0000	SEQU	FAB\$V_FTN FAB\$V_FTN=0	0
00000001	0000	0000	SEQU	FAB\$V_CR FAB\$V_CR=1	1
00000002	0000	0000	SEQU	FAB\$V_PRN FAB\$V_PRN=2	2
00000003	0000	0000	SEQU	FAB\$V_BLK FAB\$V_BLK=3	3
0000001F	0000	0000	SEQU	FAB\$B_RFM FAB\$B_RFM=31	31
00000020	0000	0000	SEQU	FAB\$L_JNL FAB\$L_JNL=32	32
00000024	0000	0000	SEQU	FAB\$L_XAB FAB\$L_XAB=36	36
00000028	0000	0000	SEQU	FAB\$L_NAM FAB\$L_NAM=40	40
0000002C	0000	0000	SEQU	FAB\$L_FNA FAB\$L_FNA=44	44
00000030	0000	0000	SEQU	FAB\$L_DNA FAB\$L_DNA=48	48
00000034	0000	0000	SEQU	FAB\$B_FNS FAB\$B_FNS=52	52

	0000	\$EQU	FAB\$B_DNS FAB\$B_DNS=53	53
00000035	0000	\$EQU	FAB\$W_MRS FAB\$W_MRS=54	54
00000036	0000	\$EQU	FAB\$L_MRN FAB\$L_MRN=56	56
00000038	0000	\$EQU	FAB\$W_BLS FAB\$W_BLS=60	60
0000003C	0000	\$EQU	FAB\$B_BKS FAB\$B_BKS=62	62
0000003E	0000	\$EQU	FAB\$B_FSZ FAB\$B_FSZ=63	63
0000003F	0000	\$EQU	FAB\$L_DEV FAB\$L_DEV=64	64
00000040	0000	\$EQU	FAB\$L_SDC FAB\$L_SDC=68	68
00000044	0000	\$EQU	FAB\$W_GBC FAB\$W_GBC=72	72
00000048	0000	\$EQU	FAB\$R_ACMODES_OVERLAY FAB\$R_ACMODES_OVERLAY=74	74
0000004A	0000	\$EQU	FAB\$B_ACMODES FAB\$B_ACMODES=74	74
0000004A	0000	\$EQU	FAB\$R_ACMODES_BITS FAB\$R_ACMODES_BITS=74	74
00000002	0000	\$EQU	FAB\$S_LNM_MODE 2 FAB\$S_LNM_MODE=2	2
00000000	0000	\$EQU	FAB\$V_LNM_MODE 0 FAB\$V_LNM_MODE=0	0
00000002	0000	\$EQU	FAB\$S_CHAN_MODE 2 FAB\$S_CHAN_MODE=2	2
00000002	0000	\$EQU	FAB\$V_CHAN_MODE 2 FAB\$V_CHAN_MODE=2	2
00000002	0000	\$EQU	FAB\$S_FILE_MODE 2 FAB\$S_FILE_MODE=2	2
00000004	0000	\$EQU	FAB\$V_FILE_MODE 4 FAB\$V_FILE_MODE=4	4
0000004B	0000	\$EQU	FAB\$R_RCF_OVERLAY FAB\$R_RCF_OVERLAY=75	75

```

0000004B 0000    $EQU   FAB$B_RCF      75
0000004B 0000    $EQU   FAB$B_RCF=75
00000000 0000    $EQU   FAB$R_RCF_BITS 75
00000000 0000    $EQU   FAB$R_RCF_BITS=75
00000000 0000    $EQU   FAB$V_RU       0
00000000 0000    $EQU   FAB$V_RU=0
00000001 0000    $EQU   FAB$V_AI       1
00000001 0000    $EQU   FAB$V_AI=1
00000002 0000    $EQU   FAB$V_BI       2
00000002 0000    $EQU   FAB$V_BI=2
00000000 0000    $DEFEND FAB,,DEF
00000000 0000    .MACRO $FABDEF A
00000000 0000    .ENDM  $FABDEF
00000000 0000    .IIF   DIF <> <GLOBAL>,DISABLE
00000000 0000    .CROSS
00000000 0000    .RESTORE
0000021C 021C    $$.TABINIT FAB$C_BID, FAB$C_BLN
0000021C 021C    .IIF NE .83, .print ;%MACRO-I=GENINFO, Generated INFO: RMS BLOCK NOT LONGWORD ALIGNE
0000021C 021C    $$.TAB=.
0000021C 021C    .BYTE FAB$C_BID
0000021C 021D    .BYTE FAB$C_BLN
0000026C 021E    .BLKB FAB$C_BLN-2
0000026C 026C    $$.TABEND=.
00000000 026C    $$.TMP=0
00000000 026C    SSR_VBFSET FAB,<>
00000000 026C    .IRP X,<>
00000000 026C    .IF DF FAB$V 'X
00000000 026C    $$.TMP=$$.TMP!<1@FAB$V_>
00000000 026C    .IFF
00000000 026C    .ERROR      ; UNDEFINED BIT VALUE CODE: X;
00000000 026C    .ENDC
00000000 026C    .ENDR
00000220 026C    .= $$ .TAB+FAB$L_FOP
00000000 0220    .ADDRESS   $$ .TMP
0000022C 0224    .= $$ .TAB+FAB$L_ALQ
00000000 022C    .ADDRESS   0
00000000 0230    .WORD     0
00000000 0232    $$.TMP=0
00000000 0232    SSR_VBFSET FAB,<GET>
00000000 0232    .IRP X,<GET>
00000000 0232    .IF DF FAB$V 'X
00000000 0232    $$.TMP=$$.TMP!<1@FAB$V_>
00000000 0232    .IFF
00000000 0232    .ERROR      ; UNDEFINED BIT VALUE CODE: X;
00000000 0232    .ENDC
00000000 0232    .ENDR
00000000 0232    .IF DF FAB$V_GET

```

- REMOTE TERMINAL PROGRAM
READ WRITE DATA

G 2

16-SEP-1984 02:15:27 VAX/VMS Macro V04-00
5-SEP-1984 03:15:47 [RTPAD.SRC]RTPAD.MAR:1

Page 50
(3)

00 0266 .BYTE <>O@FAB\$V_LNM_MODE> + <O@FAB\$V_CHAN_MODE> + -
 0267 <O@FAB\$V_FILE_MODE>
 0267 .IIF NE 8-8 .ERROR ; INVALID BYTE SIZE
 0267 .IF NB <SYSS\$INPUT>
 0267 .SAVE
 00000000 .PSECT \$RMSNAM
 00000000 \$\$.TMPX=.
 54 55 50 4E 49 24 53 59 53 0000 .ASCII %SYSS\$INPUT%
 00000009 0009 \$\$.TMPX1=-\$\$.TMPX
 00000267 .RESTORE
 00000248 0267 .= \$\$TAB+FAB\$L_FNA
 00000000' 0248 .ADDRESS \$\$TMPX
 00000250 024C .= \$\$TAB+FAB\$B_FNS
 09 0250 .BYTE \$\$TMPX1
 0251 .ENDC
 0251 .IF NB <>
 0251 .SAVE
 0251 .PSECT \$RMSNAM
 0251 \$\$.TMPX=.
 0251 .ASCII %%
 0251 \$\$.TMPX1=-\$\$.TMPX
 0251 .RESTORE
 0251 .= \$\$TAB+FAB\$L_DNA
 0251 .ADDRESS \$\$TMPX
 0251 .= \$\$TAB+FAB\$B_DNS
 0251 .BYTE \$\$TMPX1
 0251 .ENDC
 0000026C 0251 .= \$\$TABEND
 026C
 026C 802
 026C 803 SYSINRAB:: \$RAB FAB=SYSINFAB
 026C \$RABDEF
 026C \$DEFINI RAB,
 026C .SAVE LOCAL_BLOCK
 026C .NOCROSS
 026C .IIF DIF <> <GLOBAL>,.ENABLE SUPPRESSION
 026C .PSECT \$ABSS,ABS
 0000 \$GBLINI
 0000 .IF IDN <LOCAL> <GLOBAL>
 0000 .MACRO \$DEF SYM,ALLOC,SIZ
 0000 .IIF NB,SYM, SYM::
 0000 .IIF NB,ALLOC, ALLOC SIZ
 0000 .ENDM \$DEF
 0000 .MACRO \$SEQU SYM,VAL
 0000 SYM==VAL
 0000 .ENDM \$SEQU
 0000 .MACRO \$VIELD1 MOD,SEP,SYM,SIZ,MSK
 0000 SIZ...=1
 0000 .IIF NB,SIZ, SIZ...=SIZ
 0000 .IF NB,SYM
 0000 MOD'SEP'V 'SYM==BIT..
 0000 .IIF NB,SIZ, MOD'SEP'S 'SYM==SIZ
 0000 .IIF NB,MSK, MOD'SEP'M_ 'SYM==<<<1@SIZ...>-1>@BIT...>
 0000 .ENDC
 0000 BIT...=BIT...+SIZ...
 0000 .ENDM \$VIELD1
 0000 .IFF

0000 .IIF DIF <LOCAL> <LOCAL>, ERROR ;ARG MUST BE "GLOBAL","LOCAL",OR NULL
0000 .MACRO \$DEF SYM,ALLOC,SIZ
0000 .IIF NB,SYM, SYM:
0000 .IIF NB,ALLCC, ALLOC SIZ
.ENDM \$DÉF
.MACRO \$SEQU SYM,VAL
SYM=VAL
.ENDM \$SEQU
.MACRO \$VIELD1 MOD,SEP,SYM,SIZ,MSK
SIZ...=1
.IIF NB,SIZ, SIZ...=SIZ
.IF NB,SYM
MOD'SEP'V 'SYM=BIT.
.IIF NB,SIZ, MOD'SEP'S 'SYM=SIZ
.IIF NB,MSK, MOD'SEP'M_ 'SYM=<<<1@SIZ...>-1>@BIT...>
.ENDC
BIT...=BIT...+SIZ...
.ENDM \$VIELD1
.ENDC

00000000 0000 .=0

00000001 0000 \$SEQU RAB\$C_BID 1
RAB\$C_BID=1

00003FC0 0000 \$SEQU RAB\$M_PPF_RAT 16320
RAB\$M_PPF_RAT=16320

00004000 0000 \$SEQU RAB\$M_PPF_IND 16384
RAB\$M_PPF_IND=16384

00000001 0000 \$SEQU RAB\$M_ASY 1
RAB\$M_ASY=1

00000002 0000 \$SEQU RAB\$M_TPT 2
RAB\$M_TPT=2

00000004 0000 \$SEQU RAB\$M_REA 4
RAB\$M_REA=4

00000008 0000 \$SEQU RAB\$M_RRL 8
RAB\$M_RRL=8

00000010 0000 \$SEQU RAB\$M_UIF 16
RAB\$M_UIF=16

00000020 0000 \$SEQU RAB\$M_MAS 32
RAB\$M_MAS=32

00000040 0000 \$SEQU RAB\$M_FDL 64
RAB\$M_FDL=64

00000080 0000 \$SEQU RAB\$M_HSH 128
RAB\$M_HSH=128

00000100 0000 \$SEQU RAB\$M_EOF 256
RAB\$M_EOF=256

	0000	SEQU	RAB\$M_RAH RAB\$M_RAH=512	512
00000200	0000	SEQU	RAB\$M_WBH RAB\$M_WBH=1024	1024
00000400	0000	SEQU	RAB\$M_BIO RAB\$M_BIO=2048	2048
00000800	0000	SEQU	RAB\$M_LV2 RAB\$M_LV2=4096	4096
00001000	0000	SEQU	RAB\$M_LOA RAB\$M_LOA=8192	8192
00002000	0000	SEQU	RAB\$M_LIM RAB\$M_LIM=16384	16384
00004000	0000	SEQU	RAB\$M_LOC RAB\$M_LOC=65536	65536
00010000	0000	SEQU	RAB\$M_WAT RAB\$M_WAT=131072	131072
00020000	0000	SEQU	RAB\$M_ULK RAB\$M_ULK=262144	262144
00040000	0000	SEQU	RAB\$M_RLK RAB\$M_RLK=524288	524288
00080000	0000	SEQU	RAB\$M_NLK RAB\$M_NLK=1048576	1048576
00100000	0000	SEQU	RAB\$M_KGE RAB\$M_KGE=2097152	2097152
00200000	0000	SEQU	RAB\$M_KGT RAB\$M_KGT=4194304	4194304
00400000	0000	SEQU	RAB\$M_NXR RAB\$M_NXR=8388608	8388608
00800000	0000	SEQU	RAB\$M_RNE RAB\$M_RNE=16777216	16777216
01000000	0000	SEQU	RAB\$M_TMO RAB\$M_TMO=33554432	33554432
02000000	0000	SEQU	RAB\$M_CVT RAB\$M_CVT=67108864	67108864
04000000	0000	SEQU	RAB\$M_RNF RAB\$M_RNF=134217728	134217728
08000000	0000	SEQU	RAB\$METO RAB\$METO=268435456	268435456
10000000	0000			

	0000	SEQU	RAB\$M_PTA	536870912
20000000	0000		RAB\$M_PTA=536870912	
	0000	SEQU	RAB\$M_PMT	1073741824
40000000	0000		RAB\$M_PMT=1073741824	
	0000	SEQU	RAB\$M_CCO	-2147483648
80000000	0000		RAB\$M_CCO=-2147483648	
	0000	SEQU	RAB\$C_SEQ	0
00000000	0000		RAB\$C_SEQ=0	
	0000	SEQU	RAB\$C_KEY	1
00000001	0000		RAB\$C_KEY=1	
	0000	SEQU	RAB\$C_RFA	2
00000002	0000		RAB\$C_RFA=2	
	0000	SEQU	RAB\$C_STM	3
00000003	0000		RAB\$C_STM=3	
	0000	SEQU	RAB\$K_BLN	68
00000044	0000		RAB\$K_BLN=68	
	0000	SEQU	RAB\$C_BLN	68
00000044	0000		RAB\$C_BLN=68	
	0000	SEQU	RAB\$S_RABDEF	68
00000044	0000		RAB\$S_RABDEF=68	
	0000	SEQU	RAB\$B_BID	0
00000000	0000		RAB\$B_BID=0	
	0000	SEQU	RAB\$B_BLN	1
00000001	0000		RAB\$B_BLN=1	
	0000	SEQU	RAB\$R_ISI_OVERLAY	2
00000002	0000		RAB\$R_ISI_OVERLAY=2	
	0000	SEQU	RAB\$W_ISI	2
00000002	0000		RAB\$W_ISI=2	
	0000	SEQU	RAB\$R_ISI_BITS	2
00000002	0000		RAB\$R_ISI_BITS=2	
	0000	SEQU	RAB\$S_PPF_RAT	8
00000008	0000		RAB\$S_PPF_RAT=8	
	0000	SEQU	RAB\$V_PPF_RAT	6
00000006	0000		RAB\$V_PPF_RAT=6	
	0000	SEQU	RAB\$V_PPF_IND	14
0000000E	0000		RAB\$V_PPF_IND=14	
	0000	SEQU	RAB\$R_ROP_OVERLAY	4
00000004	0000		RAB\$R_ROP_OVERLAY=4	

- REMOTE TERMINAL PROGRAM
READ WRITE DATA

L 2

16-SEP-1984 02:15:27 VAX/VMS Macro V04-00
5-SEP-1984 03:15:47 [RTPAD.SRC]RTPAD.MAR;1Page 55
(3)

00000004	0000	SEQU	RAB\$L_ROP RAB\$L_ROP=4	4
00000004	0000	SEQU	RAB\$R_ROP_BITSO RAB\$R_ROP_BITSO=4	
00000000	0000	SEQU	RAB\$V_ASY RAB\$V_ASY=0	0
00000001	0000	SEQU	RAB\$V_TPT RAB\$V_TPT=1	1
00000002	0000	SEQU	RAB\$V_REA RAB\$V_REA=2	2
00000003	0000	SEQU	RAB\$V_RRL RAB\$V_RRL=3	3
00000004	0000	SEQU	RAB\$V_UIF RAB\$V_UIF=4	4
00000005	0000	SEQU	RAB\$V_MAS RAB\$V_MAS=5	5
00000006	0000	SEQU	RAB\$V_FDL RAB\$V_FDL=6	6
00000007	0000	SEQU	RAB\$V_HSH RAB\$V_HSH=7	7
00000008	0000	SEQU	RAB\$V_EOF RAB\$V_EOF=8	8
00000009	0000	SEQU	RAB\$V_RAH RAB\$V_RAH=9	9
0000000A	0000	SEQU	RAB\$V_WBH RAB\$V_WBH=10	10
0000000B	0000	SEQU	RAB\$V_BIO RAB\$V_BIO=11	11
0000000C	0000	SEQU	RAB\$V_LV2 RAB\$V_LV2=12	12
0000000D	0000	SEQU	RAB\$V_LOA RAB\$V_LOA=13	13
0000000E	0000	SEQU	RAB\$V_LIM RAB\$V_LIM=14	14
00000010	0000	SEQU	RAB\$V_LOC RAB\$V_LOC=16	16
00000011	0000	SEQU	RAB\$V_WAT RAB\$V_WAT=17	17

- REMOTE TERMINAL PROGRAM
READ WRITE DATA

M 2

16-SEP-1984 02:15:27 VAX/VMS Macro V04-00
5-SEP-1984 03:15:47 [RTPAD.SRC]RTPAD.MAR;1Page 56
(3)

	0000	SEQU	RAB\$V_ULK RAB\$V_ULK=18	18
00000012	0000	SEQU	RAB\$V_RLK RAB\$V_RLK=19	19
00000013	0000	SEQU	RAB\$V_NLK RAB\$V_NLK=20	20
00000014	0000	SEQU	RAB\$V_KGE RAB\$V_KGE=21	21
00000015	0000	SEQU	RAB\$V_KGT RAB\$V_KGT=22	22
00000016	0000	SEQU	RAB\$V_NXR RAB\$V_NXR=23	23
00000017	0000	SEQU	RAB\$V_RNE RAB\$V_RNE=24	24
00000018	0000	SEQU	RAB\$V_TMO RAB\$V_TMO=25	25
00000019	0000	SEQU	RAB\$V_CVT RAB\$V_CVT=26	26
0000001A	0000	SEQU	RAB\$V_RNF RAB\$V_RNF=27	27
0000001B	0000	SEQU	RAB\$VETO RAB\$VETO=28	28
0000001C	0000	SEQU	RAB\$V_PTA RAB\$V_PTA=29	29
0000001D	0000	SEQU	RAB\$V_PMT RAB\$V_PMT=30	30
0000001E	0000	SEQU	RAB\$V_CCO RAB\$V_CCO=31	31
00000004	0000	SEQU	RAB\$R_ROP_FIELDS RAB\$R_ROP_FIELDS=4	4
00000005	0000	SEQU	RAB\$B_ROP1 RAB\$B_ROP1=5	5
00000006	0000	SEQU	RAB\$B_ROP2 RAB\$B_ROP2=6	6
00000007	0000	SEQU	RAB\$B_ROP3 RAB\$B_ROP3=7	7
00000008	0000	SEQU	RAB\$L_STS RAB\$L_STS=8	8

- REMOTE TERMINAL PROGRAM
READ WRITE DATA

N 2

16-SEP-1984 02:15:27 VAX/VMS Macro V04-00
5-SEP-1984 03:15:47 [RTPAD.SRC]RTPAD.MAR;1Page 57
(3)

0000000C	0000	SEQU	RAB\$R_STV_OVERLAY RAB\$R_STV_OVERLAY=12	12
0000000C	0000	SEQU	RAB\$L_STV RAB\$L_STV=12	12
0000000C	0000	SEQU	RAB\$R_STV_FIELDS RAB\$R_STV_FIELDS=12	12
0000000C	0000	SEQU	RAB\$W_STV0 RAB\$W_STV0=12	12
0000000E	0000	SEQU	RAB\$W_STV2 RAB\$W_STV2=14	14
00000010	0000	SEQU	RAB\$R_RFA_OVERLAY RAB\$R_RFA_OVERLAY=16	16
00000006	0000	SEQU	RAB\$S_RFA RAB\$S_RFA=6	6
00000010	0000	SEQU	RAB\$W_RFA RAB\$W_RFA=16	16
00000010	0000	SEQU	RAB\$R_RFA_FIELDS RAB\$R_RFA_FIELDS=16	16
00000010	0000	SEQU	RAB\$L_RFA0 RAB\$L_RFA0=16	16
00000014	0000	SEQU	RAB\$W_RFA4 RAB\$W_RFA4=20	20
00000018	0000	SEQU	RAB\$L_CTX RAB\$L_CTX=24	24
0000001E	0000	SEQU	RAB\$B_RAC RAB\$B_RAC=30	30
0000001F	0000	SEQU	RAB\$B_TMO RAB\$B_TMO=31	31
00000020	0000	SEQU	RAB\$W_USZ RAB\$W_USZ=32	32
00000022	0000	SEQU	RAB\$W_RSZ RAB\$W_RSZ=34	34
00000024	0000	SEQU	RAB\$L_UBF RAB\$L_UBF=36	36
00000028	0000	SEQU	RAB\$L_RBF RAB\$L_RBF=40	40
0000002C	0000	SEQU	RAB\$L_RHB RAB\$L_RHB=44	44

	0000	SEQU	RAB\$R_KBF_OVERLAY RAB\$R_KBF_OVERLAY=48	48
00000030	0000	SEQU	RAB\$L_KBF RAB\$L_KBF=48	48
00000030	0000	SEQU	RAB\$L_PBF RAB\$L_PBF=48	48
00000034	0000	SEQU	RAB\$R_KSZ_OVERLAY RAB\$R_KSZ_OVERLAY=52	52
00000034	0000	SEQU	RAB\$B_KSZ RAB\$B_KSZ=52	52
00000034	0000	SEQU	RAB\$B_PSZ RAB\$B_PSZ=52	52
00000035	0000	SEQU	RAB\$B_KRF RAB\$B_KRF=53	53
00000036	0000	SEQU	RAB\$B_MBF RAB\$B_MBF=54	54
00000037	0000	SEQU	RAB\$B_MBC RAB\$B_MBC=55	55
00000038	0000	SEQU	RAB\$R_BKT_OVERLAY RAB\$R_BKT_OVERLAY=56	56
00000038	0000	SEQU	RAB\$L_BKT RAB\$L_BKT=56	56
00000038	0000	SEQU	RAB\$L_DCT RAB\$L_DCT=56	56
0000003C	0000	SEQU	RAB\$L_FAB RAB\$L_FAB=60	60
00000040	0000	SEQU	RAB\$L_XAB RAB\$L_XAB=64	64
			\$DEFEND RAB,,DEF .MACRO \$RABDEF A .ENDM \$RABDEF .IIF DIF <> <GLOBAL>,DISABLE .CROSS .RESTORE	SUPPRESSION
0000026C	026C		SSR TABINIT RAB\$C_BID,RAB\$C_BLN	
0000026C	026C		.IIF NE .&3, .print ;%MACRO-I=GENINFO, Generated INFO: RMS BLOCK NOT LONGWORD ALIGNE	
01	026C		\$.TAB=.	
44	026D		.BYTE RAB\$C_BID	
000002B0	026E		.BYTE RAB\$C_BLN	
			.BLKB RAB\$C_BLN-2	

000002B0 02B0 \$\$.TABEND=.

02B0

02B0 02B0 \$\$.TMP=0 SSR_VBFSET RAB,<>

02B0 .IRP X,<>

02B0 .IF DF RABSV 'X

02B0 \$\$.TMP=\$\$.TMP!<1@RABSV '_X>

02B0 .IFF

02B0 .ERROR ; UNDEFINED BIT VALUE CODE: X;

02B0 .ENDC

02B0 .ENDR

02B0

00000270 02B0 .=\$\$.TAB+RAB\$L_ROP

00000000' 0270 .ADDRESS \$\$.TMP

00000284 0274 .=\$\$.TAB+RAB\$L_CTX

00000000' 0284 .ADDRESS 0

0000028A 0288 .=\$\$.TAB+RAB\$B_RAC

028A .IF DF RAB\$C_SEQ

00 028A .BYTE RAB\$C_SEQ

028B .IFF

028B .BYTE

028B .ERROR ; UNDEFINED VALUE FOR FIELD: CNST;

028B

00 028B .BYTE 0

0000 028C .WORD 0

0000 028E .WORD 0

00000000' 0290 .ADDRESS 0

00000000' 0294 .ADDRESS 0

00000000' 0298 .ADDRESS 0

00000000' 029C 02A0 .ADDRESS 0

02A0 .IF NB <>

02A0 .=\$\$.TAB+RAB\$L_PBF

02A0 .ADDRESS

02A0 .ENDC

00 02A0 .BYTE 0

02A1 .IF NB <>

02A1 .=\$\$.TAB+RAB\$B_PSZ

02A1 .BYTE

02A1 .ENDC

00 02A1 .BYTE 0

00 02A2 .BYTE 0

00 02A3 .BYTE 0

00000000' 02A4 .ADDRESS 0

0000021C' 02A8 .ADDRESS SYSINFAB

00000000' 02AC .ADDRESS 0

000002B0 02B0 .=\$\$.TABEND

02B0

02B0 806
 02B0 807 ;
 02B0 808 ; Flags
 02B0 809 ;
 02B0 810
 00 02B0 811 INDFLAG:: .BYTE 0 ; Indicate indirect command file
 02B1 812
 00 02B1 813 WAKEFLAG:: .BYTE 0 ; Flag for legitimate \$WAKE
 02B2 814
 00000000 02B2 815 CTERM_FLAG:: .LONG 0 ; TSA/CTERM flags (see \$RTPADDEF)
 02B6 816
 00000000 02B6 817 RTLOG_FLAGS:: .LONG 0 ; value of RTPAD\$LOG
 02BA 818
 02BA 819 ; other misc. global storage
 02BA 820 ;
 02BA 821
 02BA 822 RTLOG_DESC: .LONG 16 ; RTPAD\$LOG value
 00000010 02BA 823 .LONG 16
 000002C2 02BE 824 .LONG RTLOG_BUF
 000002D2 02C2 825 RTLOG_BUF: .BLKB 16
 02D2 826
 02D2 827 RTPAD_LOGNAM: .ASCID /RTPAD\$LOG/
 24 44 41 50 54 52 000002DA'010E0000' 02D2 828 .ASCID /RTPAD\$LOG/
 47 4F 4C 02E0
 02E3 829
 00000309 02E3 830 RT\$AB_ASTBLK:: .BLKB AST\$T_BUF ; Dummy AST block
 0309 831
 00000000 0309 832 RETSTATUS:: .LONG 0 ; Save a system service status
 00000000 030D 833 QUIT_PC:: .LONG 0 ; Save PC where error happened
 0311 834
 0311 835 FIRSTCMD:: .LONG DSC\$K_CLASS_DA<DSC\$B_CLASS*8> ; First command descriptor
 02000000 0311 836 .LONG 0
 00000000 0315 837 .LONG 0
 0319 838
 00 0319 839 PROTO_ECO:: .BYTE 0 ; protocol eco level
 031A 840
 0000 031A 841 HOST_OPSYS: .WORD 0 ; host system
 031C 842
 031C 843 ;
 031C 844 ; local storage
 031C 845 ;
 031C 846
 00000000 031C 847 OLDCTRL: .LONG 0 ; CLI out of band enable flags
 0320 848
 00000000 0320 849 OLDSETRWM: .LONG 0 ; Original resource wait mode
 0324 850
 00000764 0324 851 FIRSTMSG: .BLKB AST\$T_BUF+MAXMSG ; Buffer for BIND message
 0764 852
 0000041A 0764 853 MAXMSGSIZ: .LONG MAXMSG ; Maximum message size
 0768 854

- REMOTE TERMINAL PROGRAM
PROTOCOL TABLE PSECTS

E 3

16-SEP-1984 02:15:27 VAX/VMS Macro V04-00
5-SEP-1984 03:15:47 [RTPAD.SRC]RTPAD.MAR;1Page 61
(3)

```
0768 856      .SBTTL  PROTOCOL TABLE PSECTS
00000000 857      .PSECT  PROTOTB,BYTE,NOEXE
0000 858
0000 859 PROTOTBL:
0000 860
0010 861      .WORD   <104>
00000463' 862      .LONG   CTERM_RT
0004 863      .WORD   <102>
00000000' 864      .LONG   VMSRT
000C 865
00090000 866      .PSECT  PROTOTBL,BYTE,NOEXE
0000 867
00000000 868      .PSECT  PROTOTBL1,BYTE,NOEXE
0000 869
0000 870 ENDPROTO:
0000 871
0000 872      .END    RTPAD
: <4> => CTERM protocol
: This is the initialization entry
: <2> => VMS remote terminal protocol
: This is the initialization entry
```

- REMOTE TERMINAL PROGRAM

F 3

16-SEP-1984 02:15:27 VAX/VMS Macro V04-00
5-SEP-1984 03:15:47 [RTPAD.SRC|RTPAD.MAR;1]Page 62
(3)

\$\$TAB	= 0000026C	R	03	FABSL_ALQ	= 00000010
\$\$TABEND	= 000002B0	R	03	FABSL_FNA	= 0000002C
\$\$TMP	= 00000000			FABSL_FOP	= 00000004
\$\$TMP1	= 00000001			FAB\$V_CHAN_MODE	= 00000002
\$\$TMP2	= 0000000C			FAB\$V_FILE_MODE	= 00000004
\$\$TMPX	= 00000000	R	04	FAB\$V_GET	= 00000001
\$\$TMPX1	= 00000009			FAB\$V_LNM_MODE	= 00000000
\$\$GGL	= 00000000			FABSW_GBC	= 00000048
\$\$T1	= 00000001			FINALACS	00000200 RG 03
AST\$Q_IOSB	= 00000004			FINALPATH	000001F8 RG 03
AST\$T_BUF	= 00000026			FIRSTCMD	00000311 RG 03
ASTCNT	00000044	R	03	FIRSTMSG	00000324 R 03
ASTLM	00000040	R	03	FLGSM_VAXHOST	= 00000010
BIOCNT	0000004C	R	03	FOO_RSTS_1	= 00000392 R 01
BIOIM	00000048	R	03	GETJPI_ITMI_ST	0000000C R 03
CHAR_BLOCK	00000060	RG	03	HOST_OPSYS	0000031A R 03
CLISGET_VALUE	***** X	01		INDFLAG	000002B0 RG 03
CLISPRESENT	***** X	01		INFOMSG1	000004E9 R 01
CNTRLCHAN	00000154	RG	03	INFOMSG2	00000522 R 01
CONNDESC	000001A8	R	03	INIT	000000D5 R 01
CTERM\$CLOSE_LOG	***** X	01		IOSM_CTRLYAST	= 00000080
CTERM\$OPEN_LOG	***** X	01		IOS_READVBLK	= 00000031
CTERM_FLAG	000002B2	RG	03	IOS_SETMODE	= 00000023
CTERM_RT	00000463	R	01	JPIS_ASTCNT	***** X 03
CTPSB_PRO_MSGTYPE	= 00000026			JPIS_ASTLM	***** X 03
DCS_TERM	***** X	01		JPIS_BIOCNT	***** X 03
DECNETERR	00000110	RG	03	JPIS_BIOLM	***** X 03
DEVSV_TRM	***** X	01		JPI_IOSB	00000004 R 03
DEVBUFFSIZ	00000062	R	03	LEN	00000050 R 03
DEVBUFFSIZ_TEMP	0000005C	R	03	LIB\$ASN_WTH_MBX	***** X 01
DEVCLASS	00000060	R	03	LIB\$CVT_HTB	***** X 01
DEVCLASS_TEMP	00000054	R	03	LIB\$DISABLE_CTRL	***** X 01
DEVDEPEND	00000064	R	03	LIB\$ENABLE_CTRL	***** X 01
DEVDEPEND2	00000068	R	03	LIB\$PUT_OUTPUT	***** X 01
DEVNAM	00000070	RG	03	LIB\$SIGNAL	***** X 01
DEVNAMLEN	0000006C	RG	03	LINKCHAN	00000144 RG 03
DEVTYPE	00000061	R	03	LOCAL_PID	00000000 R 03
DEVTYPE_TEMP	00000058	R	03	LOG_DESC	00000167 R 03
DIBSK_LENGTH	= 00000074			LOGFILE_DESC	000001A0 R 03
DSC\$B_CLASS	= 00000003			MAILCHAN	00000148 RG 03
DSC\$K_CLASS_D	= 00000002			MAXMSG	= 0000041A
DTE_DESC	00000172	R	03	MAXMSGSIZ	00000764 R 03
DVIS_DEVBUFSIZ	= 00000008			NODEDESC	0000017D R 03
DVIS_DEVCLASS	= 00000004			NODENAME	000001B0 RG 03
DVIS_DEVDEPEND	= 0000000A			NODE_NAME_DESC	00000198 R 03
DVIS_DEVDEPEND2	= 0000001C			NOTTERM	0000011C R 03
DVIS_DEVNAM	= 00000020			NOTVMS	00000138 R 03
DVIS_DEVTYPE	= 00000006			NOT_DTE	0000027C R 01
DVIS_UNIT	= 0000000C			OBJ_C_PREFIX	= 00000003
DVILIST	0000055E	R	01	OBJ_DESC	00000189 R 03
ENDPROTO	00000000	R	07	OLDCTRL	0000031C R 03
EXITMSG	00000128	R	03	OLDSETRWM	00000320 R 03
FAB\$B_FNS	= 00000034			OLD_DESC	0000015C R 03
FAB\$C_BID	= 00000003			PROTOTBL	00000000 R 05
FAB\$C_BLN	= 00000050			PROTO_ECO	00000319 RG 03
FAB\$C_SEQ	= 00000000			PSTHRU_CRLF	00000210 R 03
FAB\$C_VAR	= 00000002			PSTHRU_MSG	00000208 R 03

QUIT PC
 RAB\$\$_RAC
 RAB\$\$_BID
 RAB\$\$_BLN
 RAB\$\$_SEQ
 RAB\$\$_CTX
 RAB\$\$_ROP
 RDWRTCHAN
 READCHAN
 RECORD QUIT
 REM\$\$_ATPC
 REM\$\$_END
 REM\$\$_FACILITY
 REM\$\$_NETERR
 REM\$\$_NOPROT
 REM\$\$_NOTERM
 REMOTENODE
 RETSTATUS
 RTSAB_ASTBLK
 RTLOG\$V_BANNER
 RTLOG_BUF
 RTLOG_DESC
 RTLOG_FLAGS
 RTPAD
 RTPAD_LOGNAME
 SHRSK_SHRDEF
 SHRS_ATPC
 SHRS_TEXT
 SSS_EXQUOTA
 SSS_NOSUCHNODE
 SSS_NOTRAN
 SSS_WASCLR
 STR\$APPEND
 STR\$CONCAT
 STR\$FREE1_DX
 STSSK_INFO
 SYSS\$ASSIGN
 SYSS\$CANCEL
 SYSS\$CLOSE
 SYSS\$CONNECT
 SYSS\$GETDEV
 SYSS\$GETDVI
 SYSS\$GETJPIW
 SYSS\$HIBER
 SYSS\$NODE
 SYSS\$OPEN
 SYSS\$PUTMSG
 SYSS\$QIO
 SYSS\$QIOW
 SYSS\$SETAST
 SYSS\$SETRWM
 SYSS\$TRNLOG
 SYSINFAB
 SYSINRAB
 TERM\$EMULATE
 TERMCHAR
 TERMMBXCHAN

0000030D RG 03
 = 0000001E RG 03
 = 00000001
 = 00000044
 = 00000000
 = 00000018
 = 00000004
 0000014C RG 03
 0000014C RG 03
 0000049C RG 01
 = 01FE115B
 ***** X 03
 = 000001FE
 ***** X 03
 ***** X 01
 00000309 RG 03
 000002E3 RG 03
 = 00000000
 000002C2 R 03
 000002BA R 03
 000002B6 RG 03
 00000000 R 01
 000002D2 R 03
 = 00000001
 = 00001158
 = 00001130
 ***** X 01
 000004C6 R 01
 ***** GX 01
 0000021C RG 03
 0000026C RG 03
 ***** X 01
 00000084 RG 03
 00000158 RG 03

TERMUNIT
 TTYDESC
 UNS\$NET_CONNECT
 VMSRT
 WAKEFLAG
 WRITECHAN
 WRITEQIO
 00000080 RG 03
 000004D6 RG 01
 ***** X 01
 ***** X 01
 000002B1 RG 03
 00000150 RG 03
 ***** X 01

RTPAD
Psect synopsis

- REMOTE TERMINAL PROGRAM

H 3

16-SEP-1984 02:15:27 VAX/VMS Macro V04-00
5-SEP-1984 03:15:47 [RTPAD.SRC]RTPAD.MAR;1

Page 64
(3)

+-----+
! Psect synopsis !
+-----+

PSECT name

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
RTPAD	000005B6 (1462.)	01 (1.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE
\$ABSS\$	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
RTPAD	00000768 (1896.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC LONG
SRMSNAM	00000009 (9.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD WRT NOVEC BYTE
PROTOTB	0000000C (12.)	05 (5.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC BYTE
PROTOTBL	00000000 (0.)	06 (6.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC BYTE
PROTOTBL1	00000000 (0.)	07 (7.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC BYTE

+-----+
! Performance indicators !
+-----+

Phase

Phase	Page faults	CPU Time	Elapsed Time
Initialization	31	00:00:00.08	00:00:00.77
Command processing	129	00:00:00.47	00:00:04.39
Pass 1	694	00:00:17.42	00:01:07.76
Symbol table sort	31	00:00:02.85	00:00:13.32
Pass 2	578	00:00:04.24	00:00:19.02
Symbol table output	1	00:00:00.13	00:00:00.56
Psect synopsis output	0	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1466	00:00:25.22	00:01:45.84

The working set limit was 2400 pages.

173467 bytes (339 pages) of virtual memory were used to buffer the intermediate code.

There were 150 pages of symbol table space allocated to hold 2667 non-local and 46 local symbols.

872 source lines were read in Pass 1, producing 31 object records in Pass 2.

63 pages of virtual memory were used to define 56 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name

Macro library name	Macros defined
\$255\$DUA28:[RTPAD.OBJ]RTPAD.MLB;1	4
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	12
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	36
TOTALS (all libraries)	52

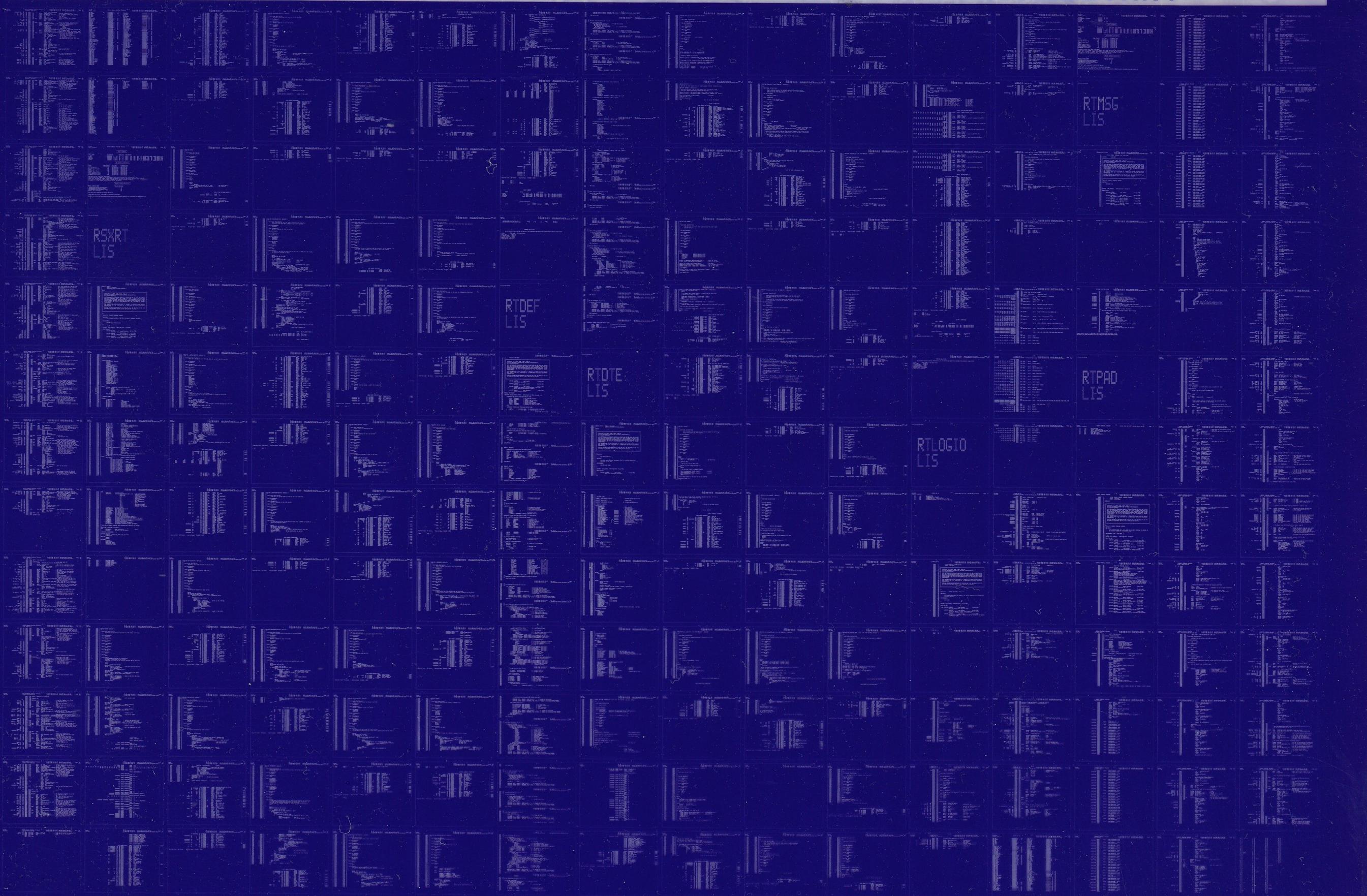
3096 GETS were required to define 52 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:\$RTPAD/OBJ=OBJ\$:\$RTPAD MSRC\$:\$RTPAD/UPDATE=(ENH\$:\$RTPAD)+EXECMLS\$:/LIB+LIB\$:\$RTPAD/LIB

0334 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0335 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

ATCODE
REQ

CLHOPS
REQ

BIGNUM REQ

BOOL
REQ

BPOSIT
REQ

BRNRTY
REQ.